The XMM-Newton ABC Guide for Hera

An Introduction To XMM-Newton Data Analysis Using the Hera Command Window

NASA/GSFC XMM-Newton Guest Observer Facility

Steve Snowden, Lynne Valencic
Brendan Perry, Michael Arida

With contributions by: Ilana Harrus, Stefan Immler, Rick Shafer, Randall Smith, Martin Still

Version 2.0

December 2013

Copies of this guide are available in `html`, `postscript` and `pdf` formats.

Contents

1 Introduction **1**
- 1.1 Acknowledgements . . . 1

2 Useful Information and References **2**
- 2.1 Main Websites . . . 2
- 2.2 *XMM-Newton* Help Desks . . . 2
- 2.3 Mission Planning and Spacecraft Status . . . 2
- 2.4 Public Data Archives . . . 3
- 2.5 Software . . . 3
- 2.6 Analysis, Documentation, and Helpful Hints . . . 3

3 A Few Words for the Veteran SAS Users **4**

4 Hera **5**
- 4.1 Placing Data on the Hera Servers . . . 5
- 4.2 Setting Up and Running Standard Hera . . . 5
- 4.3 Command Line Hera Syntax and Logic . . . 7
  - 4.3.1 Table and Command Line Syntax . . . 7
  - 4.3.2 Filtering Logic . . . 8

5 Data **9**
- 5.1 Useful Documentation . . . 9
- 5.2 The Data . . . 9
- 5.3 PI Data . . . 9
  - 5.3.1 ODF Data . . . 10
  - 5.3.2 Pipeline Product Data – Summary Files and Groupings . . . 10
  - 5.3.3 Pipeline Product Data – Data Files . . . 11

6 Preparing the Data for Processing **15**

7 An EPIC Data Processing and Analysis Primer **17**
- 7.1 Rerun the Pipeline . . . 17
- 7.2 Examine and Analyze Data . . . 18
  - 7.2.1 Create and Display an Image . . . 18
  - 7.2.2 Create and Display a Light Curve . . . 20
  - 7.2.3 Applying Standard Filters the Data . . . 20
  - 7.2.4 Applying Time Filters the Data . . . 21
  - 7.2.5 Source Detection with *edetect_chain* . . . 22
  - 7.2.6 Extract the Source and Background Spectra . . . 24
  - 7.2.7 Determine the Spectrum Extraction Areas . . . 25
  - 7.2.8 Check for Pile Up . . . 25
  - 7.2.9 My Observation is Piled Up! Now What? . . . 25
  - 7.2.10 Create the Photon Redistribution Matrix (RMF) and Ancillary File (ARF) . . . 27
  - 7.2.11 Prepare and Fit the Spectrum . . . 28
- 7.3 Timing Analysis . . . 30

8 An RGS Data Processing and Analysis Primer **33**
8.1 Rerunning the Pipeline . . . 33
8.2 Potentially useful tips for using the pipeline . . . 33
8.2.1 A Nearby Bright Optical Source . . . 33
8.2.2 A Nearby Bright X-ray Source . . . 34
8.2.3 User-defined Source Coordinates . . . 34
8.3 Examine and Analyze the Data . . . 34
8.3.1 Create and Display an Image . . . 34
8.3.2 Create and Display a Light Curve . . . 35
8.3.3 Generating the Good Time Interval (GTI) File . . . 37
8.3.4 Applying the GTI . . . 37
8.3.5 Creating the Response Matrices (RMFs) . . . 38
8.3.6 Combining Spectra . . . 38
8.4 Approaches to Spectral Fitting and the Cash Statistic . . . 38
8.4.1 Spectral Rebinning . . . 39
8.5 Fitting a Spectral Model . . . 40
8.6 Analysis of Extended Sources . . . 41
8.6.1 Region masks . . . 41
8.6.2 Making RMFs for extended sources . . . 41

9 An OM Data Processing and Analysis Primer **43**
9.1 OM Artifacts and General Information . . . 43
9.2 Imaging Mode . . . 45
9.2.1 Rerunning the Pipeline . . . 45
9.2.2 Verifying the Output . . . 45
9.3 Fast Mode . . . 46
9.3.1 Rerunning the Pipeline . . . 46
9.3.2 Verifying the Output . . . 47
9.4 Grism Analysis . . . 48
9.4.1 Rerunning the Pipeline . . . 48
9.4.2 Verifying the Output . . . 48

List of Tables

1 List of Acronyms iv

5.1 ODF data file identifiers. 12
5.2 Pipeline Processing groupings. 13
5.3 Pipeline Processing data files 14

6.1 Example datasets used in this Guide. 16

9.1 Some of the important columns in the source list file. 44
9.2 File ID equivalences between repipelined and PPS OM files. 46

Table 1: List of Acronyms

ARF	Ancillary Region File
CAL	Calibration Access Layer
CCD	Charge Coupled Device
CCF	Current Calibration File
CIF	Calibration Index File
EPIC	European Photon Imaging Camera
FITS	Flexible Image Transport System
GO	Guest Observer
GOF	NASA/GSFC Guest Observer Facility
GSFC	Goddard Space Flight Center
GUI	Graphical User Interface
HEASARC	High Energy Astrophysics Science Archive Research Center
HTML	Hyper Text Markup Language
OAL	ODF Access Layer
ODF	Observation Data File
OM	Optical Monitor
PDF	Portable Data Format
PP	Pipeline Processing System
PPS	Pipeline Processing
PV	Performance Validation
RGS	Reflection Grating Spectrometer
RMF	Redistribution Matrix File
SAS	Science Analysis System
SOC	Science Operations Center
SSC	Survey Science Centre
SV	Science Validation
XMM	X-ray Multi-Mirror Mission

Chapter 1

Introduction

The purpose of this Guide is to provide a simple walk-through of basic data extraction and analysis tasks for XMM-Newton data using SAS **as available through the Hera facility at HEASARC**. As with the original *XMM-Newton ABC Guide*, we have tried to balance providing enough information to give the user a useful introduction to a variety of analysis tasks with not providing too much information and thus overwhelming the user. This document is not intended to supercede the SAS Handbook, which is the highest authority for the use of SAS. Rather, this is meant as a general guide and introduction.

Chapters 7, 8, and 9 discuss the analysis of EPIC, RGS, and OM data respectively.

1.1 Acknowledgements

This guide would not have been possible without the help and comments from all people involved in the *X*MM-Newton project. In particular, we would like to thank Giuseppe Vacanti and Julian Osborne whose comments made this a more complete and accurate document.

IMH wishes to thank all the OM calibration team and in particular Antonio Talavera, Matteo Guainazzi and Bing Chen for their help in the preparation of this and other documents related to the OM.

SLS wishes to thank Dave Lumb, Richard Saxton, and Steve Sembay for their helpful insights into EPIC data analysis.

Chapter 2

Useful Information and References

2.1 Main Websites

- *X*MM-Newton SOC, fount of all *X*MM-Newton project information:

  `http://xmm.esac.esa.int/`

- NASA/GSFC GOF, source of US specific information and a mirror site for software and public data access:

  `http://xmm.gsfc.nasa.gov/`

- Survey Science Centre

  `http://xmmssc-www.star.le.ac.uk/`

2.2 *XMM-Newton* Help Desks

- The main project helpdesk is located at Vilspa and can be accessed through the WWW:

  `http://xmm.esac.esa.int/external/xmm_user_support/helpdesk.shtml`

  The helpdesk also provides an archive of previously asked questions.

- The NASA/GSFC GOF offers an e-mail helpdesk for both general support and for US-specific issues:

  `xmmhelp@athena.gsfc.nasa.gov`

  Some questions addressed to the NASA/GSFC GOF may be redirected to the ESA helpdesk.

2.3 Mission Planning and Spacecraft Status

- Observation Log:

  `http://xmm.esac.esa.int/external/xmm_mission_plan/index.php`

  The scheduling information from this data base has been extracted and incorporated into a Browse data base at GSFC:

  `http://heasarc.gsfc.nasa.gov/db-perl/W3Browse/w3browse.pl`

- Long-Term Timeline:

  `http://xmm.esac.esa.int/external/xmm_sched/advance_plan.shtml`

2.4 Public Data Archives

- SOC Public Data Archive via the XSA:

  http://xmm.esac.esa.int/xsa/

- GSFC Archive Mirror Site via Browse:

  http://heasarc.gsfc.nasa.gov/db-perl/W3Browse/w3browse.pl

2.5 Software

- *XMM-Newton* Standard Analysis System (SAS):

  http://xmm.esac.esa.int/sas/current/howtousesas.shtml

- HEASARC HEASoft Package:

  http://heasarc.gsfc.nasa.gov/docs/corp/software.html

- CXC CIAO Package:

  http://cxc.harvard.edu/ciao/

2.6 Analysis, Documentation, and Helpful Hints

- On-Line SAS Handbook:

  http://xmm.esac.esa.int/sas/current/doc/

- There is a “watchout” page for current SAS bugs at:

  http://xmm.esac.esa.int/sas/current/watchout/

- *XMM-Newton* Users Handbook:

  http://xmm.esac.esa.int/external/xmm_user_support/documentation/sas_usg/USG/

- The MPE Analysis Guide:

  http://www.mpe.mpg.de/xray/wave/xmm/cookbook/index.php?lang=en

- The Birmingham Analysis Guide (scripts etc. for EPIC extended source analysis):

  http://www.sr.bham.ac.uk/xmm2/

Chapter 3

A Few Words for the Veteran SAS Users

There is very little difference between processing data with Hera and SAS; that being the case, people who are already familiar with XMM-Newton data and SAS may find it useful to simply be aware of these differences, rather than go through this entire primer looking for discrepancies. For these users, the most relevant information pertains to the Hera GUI interface and placing data on the Hera servers (see §4). **Also, due to NASA network security restrictions, the pipeline task *epchain* cannot be used.** Further, the task *cifbuild* no longer needs to be run; the Hera server runs it every night, so the ccf.cif file is always up-to-date.

Users who generally prefer to run tasks from the command line may find that it is best to download their datasets from the HEASARC archive straight to their Hera account, repipeline the data there, and then analyze it using the Command Window instead of the GUI.

While tasks can also run on anonymous Hera, they take a **much** longer time, since the "files" in an anonymous user's account are really links; so when a user runs, say, *odfingest*, Hera has to replace the links with the files and unzip them before running the process.

Chapter 4

Hera

The Hera facility at HEASARC can run tasks from numerous environments, such as *CIAO*, *xselect*, and *ftools*, and includes the SAS environment, which is the software designed specifically for the reduction and analysis of XMM-Newton data. Hera lets users have access to this software remotely, without asking the users to download to the user's local machine such things as the calibration files or software. Data files stored on the user's local machine may be copied to a private space on the Hera server, or downloaded directly from the HEASARC data archive.

There are two flavors of Hera: anonymous and standard. These have identical graphical interfaces and produce identical outputs. Anonymous Hera is accessed though the HEASARC data archive by simply clicking on the "H" next to the dataset the user would like to examine. Files created through anonymous Hera will be deleted at the end of the session, but the user will be given the option to save the files to a new or existing Hera account. Standard Hera requires the user to install *fv*, available through the HEASARC, on his or her local machine (see §4.2). Standard Hera requires the user to log in to an account, and only processes data that is located on the Hera server. Files may be transferred there via anonymous Hera, HEASARC archive links, or by right-clicking the filename on the local directory list in the Hera GUI and dragging it to the desired location. Please note that only file at a time may be copied, and that due to security restrictions, whole directories cannot be copied (see §4.1).

There is no difference between the data analysis tasks in Hera's SAS and a locally-installed version of SAS – the only difference, as far as the user is concerned, is in the interface. Users who are familiar with SAS will need only to learn to navigate the Hera "wrapper"; all tasks, procedures, and techniques of data reduction are the same as if SAS were locally installed.

For the sake of simplicity, it will be assumed throughout this Guide that the user will only use SAS tasks (unless where explicitly stated). Thus, anonymous and standard Hera are synonomous with the locally-installed SAS GUI and command line.

4.1 Placing Data on the Hera Servers

Please be aware that network security regulations at NASA forbid the uploading of multiple files at once, or directories from local machines to Hera servers! Therefore, if you want to run tasks like the repipelining procedures (i.e., *emchain, emproc, epproc*), which require data from the ODF directory, you must download the ODF directory (and the PPS directory) to Hera disk space **directly from the HEASARC archive**. Users of standard Hera can click on the "Save to Hera" button in the Data Products Retrieval tab of the Browse Query Results. Users with Hera accounts will be prompted for their login information, while new users will be given the opportunity to make an account. For users of anonymous Hera, the data will be downloaded to a temporary directory for analysis. They will be given the option to save their data in an account at the session.

4.2 Setting Up and Running Standard Hera

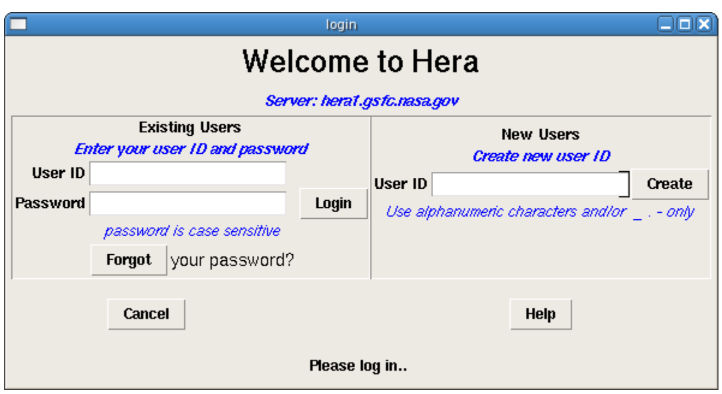
Hera currently runs on unix, Mac OS, and Windows. Unix users should install *fv* (version 4.4 or higher). Both Mac and Windows users will need to install a special Hera utility in addition to *fv*. Detailed information on downloading and installing the packages can be found at
`http://heasarc.gsfc.nasa.gov/docs/software/ftools/fv/fv_download.html`

The Hera GUI is called by either typing `fv -hera` on the command line or `fv &` on the command line and clicking on the "Connect to Hera..." option in the *fv* pop-up window, and logging in (see Figs. 4.1 and 4.2, respectively). If you do not yet have a Hera account, enter a username in the right-hand textbox and follow the instructions.

After logging in, the main Hera window will appear (see Figure 4.3). The main Hera window has five sections. The upper panel on the right side ("Remote Directory List") lists the directories and files on the NASA server in the user's account. Directly beneath it is a panel showing the directories and files on the user's local machine ("Local Directory List"). To the left are three smaller panels: the top lists any scripts the user may have ("Special Analysis Scripts"), the middle shows which packages and tasks are available ("Available Tools"), and the bottom lists which task is active, gives a short task description, and has the clickable "Help", "Run Tool...", and "Cmd Window" buttons. New accounts are created with a directory called **data** and two sample fits files, visible in the Remote Directory List window. Right-clicking on these files will bring up various options, including copying, deleting, and renaming them. Similarly, right-clicking on the data directory will bring up options including renaming the directory or creating a new one (under **data**).

Figure 4.1: The *fv* menu list.

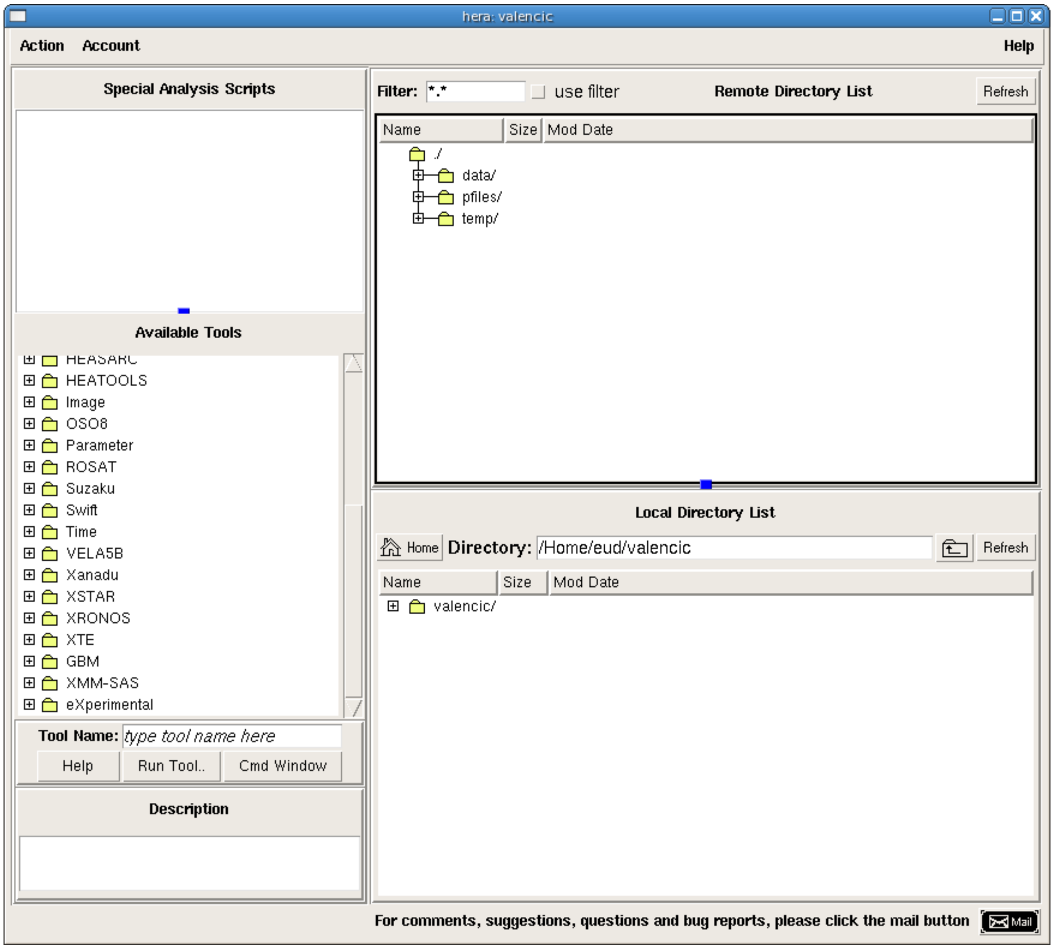
Figure 4.2: The Hera login screen.



Mission-specific software is found by single-clicking on the mission name in the middle window. SAS tasks are listed under "XMM-SAS". Once a file has been selected in the Remote Directory List panel by single-clicking it, a task can be run on it by either single-clicking on the task name, then clicking "Run Tool...", or by double-clicking the tool name. When a task is selected, its description will automatically appear in the “Description” box. Clicking on the “Help” button will pull up a website detailing the highlighted task, and clicking “Cmd Window” will cause a blank window with command line to pop up (see Fig. 4.4).

Running a task will cause a parameter window to appear, where the user can set the parameter values. When the parameters are set, clicking “Run” in the parameter window will run the task. If a Command Window is not already open, one will appear automatically, echoing the task name and all its parameter values. If Hera encounters problems while running the task, the warnings or error messages will be displayed there.

Figure 4.3: The main Hera GUI.



Command lines can often be quite long with a variety of parameters. To avoid considerable typing when creating command scripts, a feature of the standard Hera GUI interface can be of assistance: when invoking a task through the GUI, a copy of the full command appears in the Command Window; this can quickly be cut and pasted into a text file for future use.

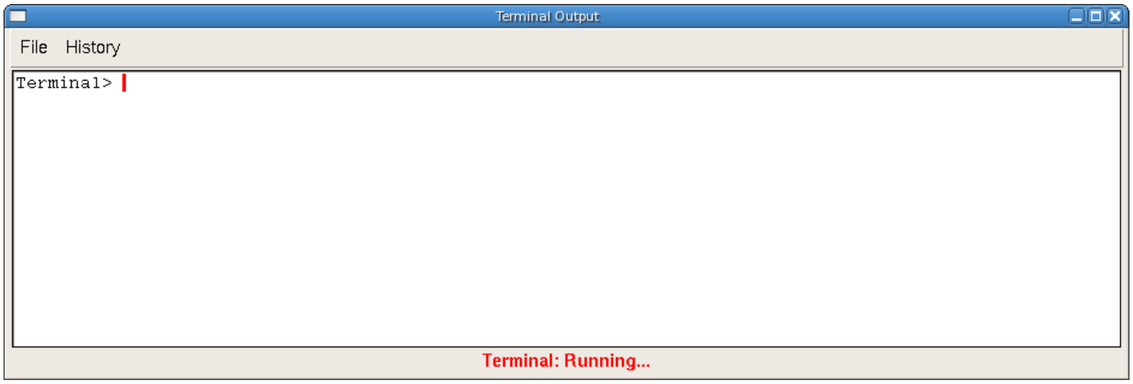
4.3 Command Line Hera Syntax and Logic

4.3.1 Table and Command Line Syntax

Veteran command-line SAS users, take note: there is some flexibility in command line Hera syntax, but less than what is in locally-installed SAS! First, all input tables in Hera are assumed to be event tables. Thus, there is no need to specify the EVENTS table when opening a file. The valid commands for opening a file in the Command Window are, therefore,

```
evselect table=filtered.fits
evselect table='filtered.fits'
```

Figure 4.4: The Command Window.




```
evselect table="filtered.fits"
```

The following syntaxes, while valid in the local installation of SAS, **are not valid** in Hera!

```
evselect table=filtered.fits:EVENTS
evselect table='filtered.fits:EVENTS'
evselect table="filtered.fits:EVENTS"
evselect table=filtered.fits%EVENTS
```

Further, the following are all valid task calls on the command line that result in identical operations:

```
evselect updateexposure=T
evselect updateexposure=yes
evselect updateexposure='yes'
evselect updateexposure="yes"
```

However,

```
evselect --updateexposure=T
evselect --updateexposure=yes
evselect --updateexposure='yes'
evselect --updateexposure="yes"
evselect -updateexposure=T
evselect -updateexposure=yes
evselect -updateexposure='yes'
evselect -updateexposure="yes"
```

are not correct syntax.

One format is not "more correct" than another, and the choice of which to use is left to user preference. In this guide we adopt the simplest format, and use no dashes and only single quotation marks only when required, such in denoting a list or filter, e.g.,

```
evselect filtertype=expression expression='(X,Y) in CIRCLE(26920,26400,500)'
```

4.3.2 Filtering Logic

Filtering event files requires some command of the SAS logical language which consists of familiar arithmetic and Boolean operators and functions. A full description can be found at the XMM-Newton SOC website, and will be explained in this Guide as they are used.

Chapter 5

Data

5.1 Useful Documentation

The documents most likely to be useful to the Hera user are those pertaining to SAS. There are a number of these which users of XMM-Newton data should be aware of. These documents include the *SSC Products Specification*, *Data Files Handbook*, *Reading Data Products CD's* (the most recent versions of these documents can be found in the SOC Document section under
`http://xmm.esac.esa.int/external/xmm_user_support/documentation/index.shtml`), and the *SAS Users Guide* (`http://xmm.esac.esa.int/external/xmm_user_support/documentation/sas_usg/USG/`).

Additional information concerning *X*MM-Newton data files can be found in the *I*nterface Control Document: Observation and Slew Data Files (XSCS to SSC) (SciSIM to SOCSIM) (XMM-SOC-ICD-0004-SSD). This is an impressive tome which goes into great detail about the file nomenclature and structure. This document can be found in the documents area of the SOC web pages:
`http://xmm.esac.esa.int/`.

5.2 The Data

One of the first steps that should be taken when examining your data is to check to see what you actually have. XMM-Newton observations can be broken into several exposures which are each assigned separate observation numbers. These separate exposures can be radically different in length and can also have the different instruments in different modes. For example, in one case the full observation was 60 ks with EPIC and RGS active but there was one delivered exposure which was ~ 3 ks and had only RGS active. (This can happen because the RGS can operate farther into regions of higher radiation than the EPIC detector. The additional observation time can be considered an additional exposure with only the RGS active.) Two files are useful and should be downloaded to the user's machine for this examination. First, the primary HTML page is `INDEX.HTM` which is included in the Pipeline Products. This page lists basic information for the observation plus the operational modes, filters, and exposure start and stop times for the individual instruments. It also has links to various summary pages, including those for the instruments. (In the case above, the EPIC summary page simply stated that "EPIC exposures processed by PPS None.") Specifically, **LOOK** at the `P*SUMMAR0000.HTM` files in the pipeline products (easily available through the links). Second, to quickly access images from the various instruments examine the PPSGRA Pipeline Products page (§ 5.3.3).

5.3 PI Data

Proprietary XMM-Newton data is available for download via your XSA account. Email instructions from the XMM-SOC at Vilspa are sent to the address on record with detailed directions on how to retrieve your data via the XSA.

XMM-Newton data is available for download at the HEASARC data archive. The data files can be considered to come in two groups in separate subdirectories when retrieved, the Observation Data Files (ODF) files and Pipeline Processing (PPS) files. The ODF data contain all of the observation-specific data necessary for reprocessing the observation. The PPS data contain, among other things, calibrated photon event files and source lists.

5.3.1 ODF Data

ODF data come with file names in the following format:

- `mmmm_iiiiiijjkk_aabeeeccfff.zzz`

  `mmmm` – revolution orbit number

  `iiiiii` – proposal number

  `jj` – target ID number in proposal

  `kk` – exposure number for target

  (**NOTE:** The ten-digit combination of `iiiiiijjkk` is the **observation number** and is used repetitively throughout the file nomenclature.)

  `aa` – detector (`M1` – MOS1, `M2` – MOS2, `PN` – PN, `OM` – OM, `R1` – RGS1, `R2` – RGS2, `SC` – spacecraft)

  `b` – flag for scheduled (`S`) or unscheduled (`U`) observations, or `X` for general purpose files)

  `eee` – exposure number

  `cc` – CCD number or OM window number

  `fff` – data identifier for the three detectors or spacecraft itself; see Table 5.1.

  `zzz` – format (`FIT` - FITS, `ASC` - ASCII)

5.3.2 Pipeline Product Data – Summary Files and Groupings

The Pipeline Processing (PP) produces quite a number of useful products which allow a first look at the data, but can overwhelm the user by their sheer numbers. The first place to look is the `INDEX.HTM` page which organizes the presentation of the data and provides links to other PP pages. The `INDEX.HTM` page also lists general observation information (target, date, time, etc.) and instrument modes.

The `INDEX.HTM` page provides links to various observation summary pages, which have names with the following nomenclature:

- `PPiiiiiijjkkAAX000SUMMAR0000.HTM`

  `iiiiiijjkk` – observation number

  `AA` – detector ID (`EP` - EPIC, `OM` - Optical Monitor, `RG` - RGS, `OB` - Observation)

PP data contain some immediately useful data products such as calibrated photon event lists, source lists, and images. While there are a large number of products which come in a single directory, they can be associated in up to 15 groupings; see Table 5.2. (The number of groups can vary depending on the number of operational instruments, e.g., if the OM is turned off there are no OM products.) Further information on each of these groupings and associated files, such as file contents, file types, and how they may be viewed, can be found in Table 5.3. Each group has an associated HTML file which organizes access to the files and provides a limited description of them. The names of the HTML files are of the following form:

- `PPiiiiiijjkkAAAAAA000_0.HTM`

  `iiiiiijjkk` – observation number

  `AAAAAA` – group identifier (see Table 5.2)

5.3.3 Pipeline Product Data – Data Files

The data file names are of the form (see Table 41 in the *XMM Data Files Handbook*):

- `PiiiiiijjkkaablllCCCCCCnmmm.zzz`

  `iiiiiijjkk` – observation number

  `aa` – detector, `M1` - MOS1, `M2` - MOS2, `PN` - PN, `CA` - for files from the CRSCOR group, `R1` - RGS1, `R2` - RGS2, `OM` - OM.

  `b` – flag for scheduled (`S`) or unscheduled (`U`) observations, or `X` for files from the CRSCOR group (and any product that is not due to a single exposure)

  `lll` – exposure number

  `CCCCCC` – file identification for data from each detector; see Table 5.3

  `n` – For EPIC data, this is the exposure map band number; for RGS data, this is the spectral order number; for the OM, this is the OM window within the exposure.

  `mmm` – source number in hexidecimal

  `zzz` – file type (e.g., PDF, PNG, FTZ, HTM)

  `ASC` - ASCII file, use a web browser, or the "more" command
  `ASZ` - gzipped ASCII file
  `FTZ` - gzipped FITS format, use *ds9, Ximage, Xselect, fv*
  `HTM` - HTML file, use Firefox or other web browser
  `PDF` - Portable Data Format, use *Acrobat Reader*
  `PNG` - Portable Networks Graphics file, use a web browser
  `TAR` - TAR file

Table 5.1: ODF data file identifiers.

Data ID	Contents
	EPIC files
IME	Event list for individual CCDs, imaging mode
RIE	Event list for individual CCDs, reduced imaging mode
CTE	Event list for individual CCDs, compressed timing mode
TIE	Event list for individual CCDs, timing mode
BUE	Event list for individual CCDs, burst mode
AUX	Auxiliary file
CCX	Counting cycle report (auxiliary file)
HBH	HBR buffer size, non-periodic housekeeping
HCH	HBR configuration, non-periodic housekeeping
HTH	HBR threshold values, non-periodic housekeeping
PEH	Periodic housekeeping
PTH	Bright pixel table, non-periodic housekeeping
DLI	Discarded lines data
PAH	Additional periodic housekeeping
PMH	Main periodic housekeeping
	RGS files
AUX	on-board processing statistics
SPE	raw event list for one CCD
DII	diagnostic images
D1H	CCD readout settings
D2H	CCD readout settings
PFH	housekeeping data
ODX	pixel offset data
	XMM files
ATS	spacecraft attitude history
	OM files
IMI	imaging file
THX	tracking history file
WDX	window data auxiliary file
NPH	non-periodic housekeeping file
PEH	periodic housekeeping file
PAX	field acquisition data
RFX	priority reference frame data
PFX	priority fast mode data
FAE	event list (if fast mode was used)

Table 5.2: Pipeline Processing groupings.

Group ID	Contents
	PP files
PPSDAT	Contains the Calibration Index File (CIF) used in the pipeline processing (*CALIND*), PPS information, and the attitude history time series (*ATTTSR*) in gzipped FITS or ASCII format.
PPSGRA	Contains the OM tracking history plots, PPS, EPIC, OM, RGS observation, and PPS run summaries. **NOTE: CHECK THESE OUT**
PPSMSG	ASCII file containing pipeline processing report
	EPIC files
CRSCOR	Contains PDF files of POSS II finding charts, HTML files of cross correlations with the SIMBAD data base, FITS tables for the detected sources
EANCIL	Contains the exposure maps in a variety of energy bands and the source-detection sensitivity maps for the EPIC instruments. The sensitivities are in units of counts s^{-1} corrected for vignetting and corresponding to a likelihood specified in the FITS header. The files are gzipped with a .FTZ extension.
EEVLIS	Contains calibrated photon event files for the EPIC detectors. If the files are sufficiently large they may be separated into two tar files. The files are gzipped fits files with a .FTZ extension.
ESKYIM	This group contains the event images in a variety of energy bands. The fits files are gzipped with a .FTZ extension, the full images also come as PNG images.
ESRLIS	Contains EPIC observation source lists. There is an HTML page of the merged source list and gzipped fits tables of source lists from the different instruments and source detection tasks.
	OM files
OIMAGE	Contains OM sky images in gzipped FITS format.
OMSLIS	Contains OM observation source lists in gzipped FITS format.
OMSRTS	Contains OM star tracking time series in gzipped FITS format.
	RGS files
REVLIS	Contains the RGS source and event lists in gzipped FITS format
REXPIM	Contains the RGS exposure maps in gzipped FITS format
RIMAGE	Contains the RGS images (both energy dispersion and cross dispersion) in gzipped FITS and PNG formats
RSPECT	Contains the RGS source and background spectra in gzipped FITS and PDF formats

Table 5.3: Pipeline Processing data files

Group ID	File ID	Contents	File Type	View With
		EPIC files		
CRSCOR	FCHART	Finding chart	PDF	*Acrobat Reader*
	ROSIMG	ROSAT image of region	PDF	*Acrobat Reader*
	SNNNNN[1]	Source cross-correlation results	Zipped FITS	*fv*
	DNNNNN[1]	Catalog descriptions	PDF	*Acrobat Reader*
	FNNNNN[1]	FOV cross-correlation result	Zipped FITS	*fv*
ESKYIM	IMAGE_8	Sky image 0.2 - 12.0 keV	Zipped FITS	*ds9, Ximage, fv*
	IMAGE_1	Sky image 0.2 - 0.5 keV	Zipped FITS	*ds9, Ximage, fv*
	IMAGE_2	Sky image 0.5 - 2.0 keV	Zipped FITS	*ds9, Ximage, fv*
	IMAGE_3	Sky image 2.0 - 4.5 keV	Zipped FITS	*ds9, Ximage, fv*
	IMAGE_4	Sky image 4.5 - 7.5 keV	Zipped FITS	*ds9, Ximage, fv*
	IMAGE_5	Sky image 7.5 - 12.0 keV	Zipped FITS	*ds9, Ximage, fv*
EANCIL	EXPMAP_8	Exposure map 0.2 - 12.0 keV	Zipped FITS, PNG	*ds9, Ximage, fv*, web browser
	EXPMAP_1	Exposure map 0.2 - 0.5 keV	Zipped FITS	*ds9, Ximage, fv*
	EXPMAP_2	Exposure map 0.5 - 2.0 keV	Zipped FITS	*ds9, Ximage, fv*
	EXPMAP_3	Exposure map 2.0 - 4.5 keV	Zipped FITS	*ds9, Ximage, fv*
	EXPMAP_4	Exposure map 4.5 - 7.5 keV	Zipped FITS	*ds9, Ximage, fv*
	EXPMAP_5	Exposure map 7.5 - 12.0 keV	Zipped FITS	*ds9, Ximage, fv*
EEVLIS[2]	MIEVLI	MOS imaging mode event list	Zipped FITS	*xmmselect, fv, Xselect*
	PIEVLI	PN imaging mode event list	Zipped FITS	*xmmselect, fv, Xselect*
	TIEVLI	PN, MOS timing mode event list	Zipped FITS	*xmmselect, fv, Xselect*
ESRLIS	EBLSLI	Box-local detect source list	Zipped FITS	*fv*
	EBMSLI	Box-map detect source list	Zipped FITS	*fv*
	EMSRLI	Max-like detect source list	Zipped FITS	*fv*
	OBSMLI	Summary source list	Zipped FITS, HTML	*fv*, web browser
		RGS files		
REVLIS	SRCLI_	RGS Source Lists	Zipped FITS	*fv*
	EVENLI	RGS Event lists	Zipped FITS	*xmmselect, fv*
REXPIM	EXPMAP	RGS Exposure Maps	Zipped FITS	*ds9, Ximage, fv*
RSPECT	SRSPEC1	1st Order Source Spectra	Zipped FITS	*Xspec, fv*
	SRSPEC2	2nd Order Source Spectra	Zipped FITS	*Xspec, fv*
	BGSPEC1	1st Order Source Spectra	Zipped FITS	*Xspec, fv*
	BGSPEC2	2nd Order Source Spectra	Zipped FITS	*Xspec, fv*
	SRSPEC	Spectra Plots	PDF format	*Acrobat Reader*
RIMAGE	ORDIMG	Images, disp. vs. X-disp	Zipped FITS, PNG	*ds9, Ximage, fv*, web browser
	IMAGE_	Images, disp. vs. PI	Zipped FITS, PNG	*ds9, Ximage, fv*, web browser
		OM files		
OANCIL	SWSREG	OM Region File	ASCII text	text editor
OIMAGE	SIMAGE	OM Sky Image	Gzipped FITS	*ds9, Ximage, fv*
OMSLIS	SWSRLI	OM Source Lists	Zipped FITS	*fv*
OMSRTS	TSTRTS	Tracking Star Time Series	Zipped FITS	*fv*

[1] NNNNN - Alphanumeric ID
[2] Files for only those modes which were active will be included

Chapter 6

Preparing the Data for Processing

Throughout this Primer, it will be assumed that the user wants to perform all tasks through the Command Window. For instructions on reducing data with the GUI, see the ABC Guide to Data Reduction with the Hera GUI.

In the sections that follow, data from the HEASARC archive are used to illustrate how to run tasks for each instrument on XMM-Newton; new users are encouraged to use these sample data, though in should be noted that any data from the relevant instrument will do. Information about the example datasets are in Table 6.1.

Regardless of which instrument you are interested in, one task is necessary to prepare the data for processing, *odfingest*. To demonstrate *odfingest*, and a host of other tasks in §7, we will use the Lockman Hole data with ObsID 0123700101, though any dataset will work equally well.

To begin, either run anonymous Hera by clicking on the "H" next to the dataset in the archive (see §4), or download the dataset from the HEASARC archive and initialize standard Hera (see §4.1 and §4.2, respectively).

Select all the zipped files in the ODF directory and unzip them; multiple files can be selected at once by holding the shift or control key, and right-clicking on the selected files will bring up a menu with the option to unzip them. (This takes a long time.)

The task *odfingest* extends the Observation Data File (ODF) summary file with data extracted from the instrument housekeeping data files and the calibration database. It is required for reprocessing the ODF data with the pipeline tasks, as well as for many other tasks, and it is only necessary to run it once on any dataset. If for some reason *odfingest* must be rerun, you must first delete the earlier file it produced. This file largely follows the naming convention described in §5.3.3, but has `SUM.SAS` appended to it. Hera will automatically set the environment parameter `SAS_ODF` to the `SUM.SAS` file in the active dataset's ODF directory, if one exists. If there is no `SUM.SAS` file, the `SAS_ODF` variable maintains its default value (the dataset's ODF directory).

To run *odfingest*, first make sure that the ODF directory is highlighted in the GUI, then click 'Cmd Window'. Due to network security issues, you cannot use the standard linux commands to navigate between directories, move files, etc. The only way to be certain that your Command Window is in the correct directory is to highlight the directory it in the GUI, then call up a new Command Window.
In the Command Window, enter

```
odfingest
```

The command and parameters will be echoed and dialog from the task will be shown as in a normal terminal. The warnings can be ignored. The prompt will return when it is complete.

At this point, the data is ready to be repipelined and analyzed. Data from the EPIC camera is discussed in §7, the RGS is in §8, and the OM is in §9. It is strongly recommended that all reprocessed data be kept in its own directory, and all the following chapters will assume that tasks are being called from the PROC directory.
To make PROC, select and right-click on the ObsID directory; the option to create a new directory will be in the menu. Select PROC and call up a new Command Window. (The old one can be closed.) Please note that new directories will always be placed in the directory that was highlighted when the command was given, and

Table 6.1: Example datasets used in this Guide.

Instrument	Chapter	ObsID	Object
EPIC	7	0123700101	Lockman Hole
EPIC (Timing mode)	7.3	0122700101	G21.5-09
RGS	8	0153950701	Mkn 421
OM (Image mode)	9.2	0123700101	Lockman Hole
OM (Fast mode)	9.3	0411081601	Mkn 421
OM (Grism mode)	9.4	0125320801	BPM 16274

similarly, all processing and analysis tasks that are called will be run from (and place output in) whichever directory is highlighted. So if you flip between directories often, it is a good idea to verify you are where you want to be before calling a task.

Chapter 7

An EPIC Data Processing and Analysis Primer

While a variety of analysis packages can be used for the following steps, the SAS was designed for the basic reduction and analysis of XMM-Newton data (extraction of spatial, spectral, and temporal data); therefore, it will be used here for demonstration purposes.

NOTE: For PN observations with very bright sources, out-of-time events can provide a serious contamination of the image. Out-of-time events occur because the read-out period for the CCDs can be up to $\sim 6.3\%$ of the frame time. Since events that occur during the read-out period can't be distinguished from others events, they are included in the event files but have invalid locations. For observations with bright sources, this can cause bright stripes in the image along the CCD read-out direction.

At this point, it is assumed that you have downloaded the data from the HEASARC archive onto a Hera server, standard or anonymous Hera is running (see §4.2), and you have prepared the data for processing (see §6). Throughout this chapter, as in §6, we will use the Lockman Hole dataset with ObsID 0123700101, though any dataset will suffice.

7.1 Rerun the Pipeline

If a dataset is more than a year old, it was probably processed with older versions of CCF and SAS prior to archiving, so the pipeline should be rerun to generate event files with the latest calibrations. The MOS has two pipeline tasks, *emchain* and *emproc*, while the PN has one, *epproc* for the PN. The two MOS tasks produce the same output, so which one to use is entirely a matter of the user's personal preference.

Verify that the working directory PROC is highlighted in the GUI. In the new Command Window you made at the end of §6, run the task(s):

```
emchain
```

or

```
emproc
```

and

```
epproc
```

If the dataset has more than one exposure, a specific exposure can be accessed using the `exposure` parameter, e.g.:

```
emchain exposure=n
```

where n is the exposure number. To create an out-of-time event file for your PN data, add the parameter *withoutoftime* to your *epproc* invocation:

```
epproc withoutoftime=yes
```

By default, none of these tasks keep any intermediate files they generate. *Emchain* maintains the naming convention described in §5.3.3. *Emproc* and *epproc* designate their output event files with "Evts.ds"; "*ImagingEvts.ds", "*TimingEvts.ds", and "*BurstEvts.ds" denote the imaging mode, timing, and burst mode event lists, respectively. In either case, you may want to name the new files something easy to type. Right-clicking on a file will give you the option to rename it.

Once the new event files have been obtained, the analysis techniques described below can be used. We will refer to the new event files as mos1.fits, mos2.fits, and pn.fits.

7.2 Examine and Analyze Data

7.2.1 Create and Display an Image

To create an image in sky coordinates, type

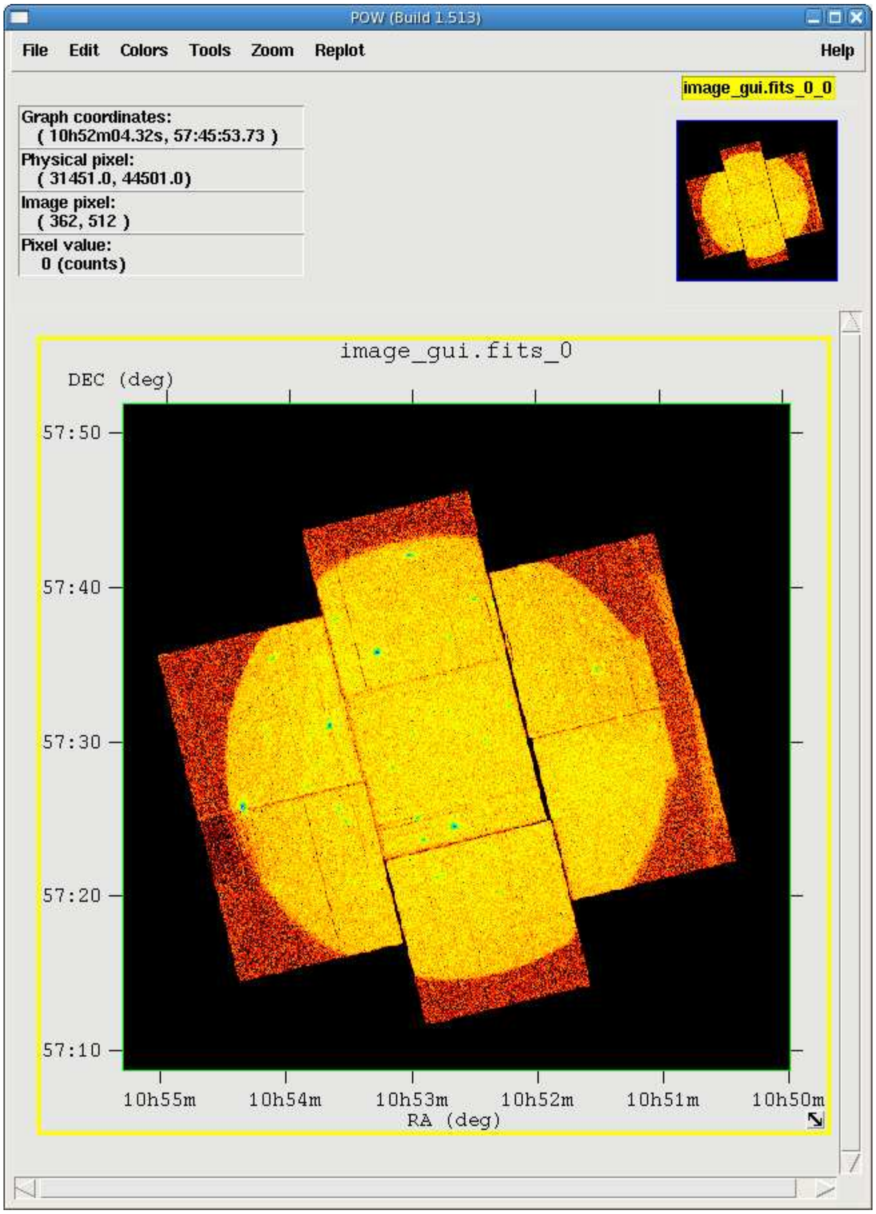
```
evselect table=mos1.fits withimageset=yes imageset=image.fits
   xcolumn=X ycolumn=Y imagebinning=imageSize ximagesize=600 yimagesize=600
```

where

`table` – input event table
`withimageset` – make an image
`imageset` – name of output image
`xcolumn` – event column for X axis
`ycolumn` – event column for Y axis
`imagebinning` – form of binning, force entire image into a given size or bin by a specified number of pixels
`ximagesize` – output image pixels in X
`yimagesize` – output image pixels in Y

The resultant image is written to the file `image.fits`. It can be viewed with *POW*, or downloaded to your local machine and viewed with *ds9*; see Figure 7.1.

Figure 7.1: The MOS1 image, displayed in *fv*.



7.2.2 Create and Display a Light Curve

To create a light curve, type

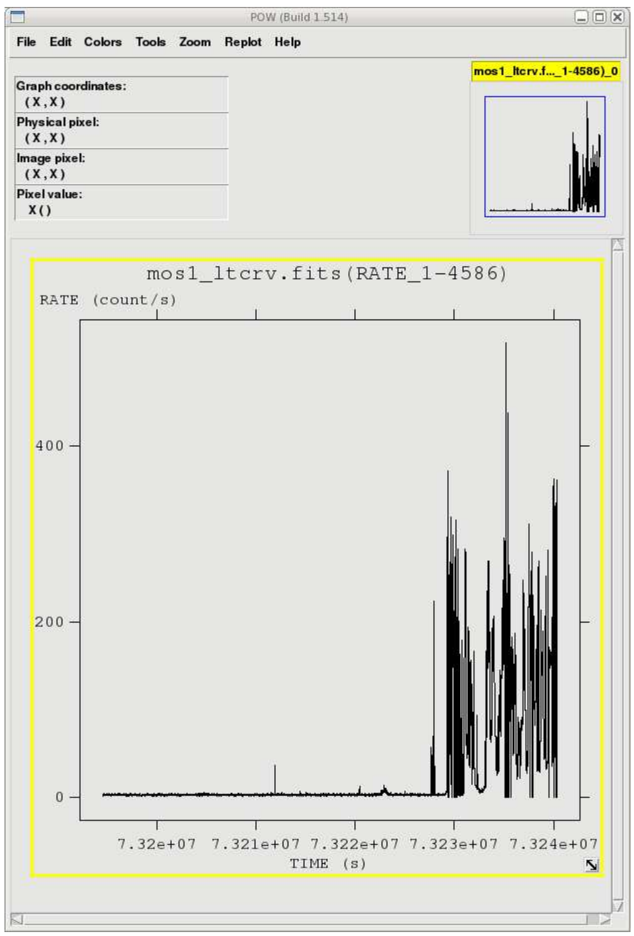
```
evselect table=mos1.fits withrateset=yes rateset=mos1_ltcrv.fits
   maketimecolumn=yes timecolumn=TIME timebinsize=100 makeratecolumn=yes
```

where

`table` – input event table
`withrateset` – make a light curve
`rateset` – name of output light curve file
`maketimecolumn` – control to create a time column
`timecolumn` – time column label
`timebinsize` – time binning (seconds)
`makeratecolumn` – control to create a count rate column, otherwise a count column will be created

The output file `mos1_ltcrv.fits` can be viewed with *POW*; see Fig. 7.2.

Figure 7.2: The light curve.



7.2.3 Applying Standard Filters the Data

The filtering expressions for the MOS and PN are:

```
(PATTERN <= 12)&&(PI in [200:12000])&&#XMMEA_EM
```

and

```
(PATTERN <= 12)&&(PI in [200:15000])&&#XMMEA_EP
```

If the PN data is timed, then the PATTERN parameter should be set to 4:

```
(PATTERN <= 4)&&(PI in [200:15000])&&#XMMEA_EP.
```

The first two expressions will select good events with PATTERN in the 0 to 12 range, and the last will select events with PATTERN between 0 and 4. The PATTERN value is similar the GRADE selection for *ASCA* data, and is related to the number and pattern of the CCD pixels triggered for a given event.The PATTERN assignments are: single pixel events: `PATTERN == 0`, double pixel events: `PATTERN in [1:4]`, triple and quadruple events: `PATTERN in [5:12]`.

The second keyword in the expressions, PI, selects the preferred pulse height of the event; for the MOS, this should be between 200 and 12000 eV. For the PN, this should be between 200 and 15000 eV. This should clean up the image significantly with most of the rest of the obvious contamination due to low pulse height events. Setting the lower PI channel limit somewhat higher (e.g., to 300 eV) will eliminate much of the rest.

Finally, the #XMMEA_EM (#XMMEA_EP for the PN) filter provides a canned screening set of FLAG values for the event. (The FLAG value provides a bit encoding of various event conditions, e.g., near hot pixels or outside of the field of view.) Setting `FLAG == 0` in the selection expression provides the most conservative screening criteria and should always be used when serious spectral analysis is to be done on the PN.

It is a good idea to keep the output filtered event files and use them in your analyses, as opposed to re-filtering the original file with every task. This will save much time and computer memory. As an example, the Lockman Hole data's original event file is 48.4 Mb; the fully filtered list (that is, filtered spatially, temporally, and spectrally) is only 4.0Mb!

To filter the data, type

```
evselect table=mos1.fits withfilteredset=yes
   expression='(PATTERN <= 12)&&(PI in [200:12000])&&#XMMEA_EM'
   filteredset=mos1_filt.fits filtertype=expression keepfilteroutput=yes
   updateexposure=yes filterexposure=yes
```

where

`table` – input event table
`filtertype` – method of filtering
`expression` – filtering expression
`withfilteredset` – create a filtered set
`filteredset` – output file name
`keepfilteroutput` – save the filtered output
`updateexposure` – for use with temporal filtering
`filterexposure` – for use with temporal filtering

7.2.4 Applying Time Filters the Data

Sometimes, it is necessary to use filters on time in addition to those mentioned above. This is because of soft proton background flaring, which can have count rates of 100 counts/sec or higher.

It should be noted that the amount of flaring that needs to be removed depends in part on the object observed; a faint, extended object will be more affected than a very bright X-ray source.

There are two ways to filter on time: with an explicit reference to the TIME or RATE parameters in the filtering expression, or by creating a secondary Good Time Interval (GTI) file with the task *tabgtigen*. Both procedures are described below. For the example data, we will filter by time, though you can just as easily filter by rate.

To explicitly define the TIME or RATE parameters, make a light curve and display it, as demonstrated in §7.2.2 and plotted in Figure 7.2. There is a very large flare toward the end of the observation, so the syntax for the time selection is `(TIME <= 7.32273e7)`. However, there is also a small flare within an otherwise good interval. A slightly more comlicated expression to remove it would be: `(TIME <= 7.32273e7)&&!(TIME IN [7.32219e7:7.32238e7])`. The syntax `&&(TIME < 7.32273e7)` includes only events with times less than 7.32273e7, and the "!" symbol stands for the logical "not", so use `&&!(TIME in [7.32219e7:7.32238e7])` to exclude events in the time interval 7.32219e7 to 7.32238e7.

If combined with the standard filtering expression (see §7.2.3), the full filtering expression would then be:

```
(PATTERN <= 12)&&(PI in [200:12000])&&#XMMEA_EM
   &&(TIME <= 7.32273e7) &&!(TIME in [7.32219e7:7.32238e7])
```

This expression can then be used to filter the original event file, as shown in §7.2.3, or only the times can be used to filter the file that has already had the standard filters applied:

```
evselect table=mos1_filt.fits withfilteredset=yes
   expression='(TIME <= 7.32273e7) &&!(TIME in [7.32219e7:7.32238e7]) '
   filteredset=mos1_filt_time.fits filtertype=expression keepfilteroutput=yes
   updateexposure=yes filterexposure=yes
```

where the keywords are as described in §7.2.3.

To filter on time using a secondary GTI file, make the file by using the same time filtering parameters as determined above and the `tabgtigen` task,

```
tabgtigen table=mos1_ltcrv.fits gtiset=gtiset.fits timecolumn=TIME
   expression='(TIME <= 7.32273e7) &&!(TIME in [7.32219e7:7.32238e7])'
```

where

`table` – input count rate table
`expression` – filtering expression
`gtiset` – output file name for selected GTI intervals
`timecolumn` – time column

and apply the new GTI file with the `evselect` task:

```
evselect table=mos1_filt.fits withfilteredset=yes
   expression='GTI(gtiset.fits,TIME)' filteredset=mos1_filt_time.fits
   filtertype=expression keepfilteroutput=yes
   updateexposure=yes filterexposure=yes
```

where

`table` – input count rate table
`expression` – filtering expression
`withfilteredset` – create a filterered set
`filteredset` – output file name
`filtertype` – method of filtering
`keepfilteroutput` – save the filtered set
`updateexposure` – update exposure information in event list and in spectrum files
`filterexposure` – filter exposure extensions of event list with same time
  filters as for corresponding CCD

7.2.5 Source Detection with *edetect_chain*

The task *edetect_chain* is a metatask that does nearly all the work involved with EPIC source detection. It can take as input arbitrary combinations of images from different energy bands and different EPIC instruments, with up to three instruments and 5 energy bands. However, users should be aware that that the binning and WCS keywords in all images **must be identical**. *Edetect_chain* is comprised of seven straightforward tasks that can also be run by hand. *Edetect_chain* requires input files to be generated and prepared using the tasks *atthkgen* and *evselect*; the task *emosaic*, while not necessary for source detection, does provide a nice mosaicked image for display purposes. Fortunately, these are all quick and straightforward.

In the example below, source detection on images is done in two bands, 500 - 1000 eV and 4500 - 12000 eV, which correspond to bands 2 and 5 of the 3XMM Catalogue, for all three detectors. The source count rates are converted into fluxes through the energy converstion factors (ECFs) for each detector and energy band. The ECFs depend on the pattern selection and filter used during the observation and are given in units of 10^{11} cts cm^2/erg. Interested users can find all of the bands listed in Table 1 of the Catalogue. The ECFs are in Table 8 of the Catalogue.)
The example uses the filtered event files from all three cameras, which can be produced in §7.2.4.

First, make the attitude file:

```
atthkgen atthkset=attitude.fits timestep=1
```

where

`atthkset` – output file name
`timestep` – time step in seconds for attitude file

Next, make the band 2 and band 5 images with *evselect*. We'll start with the band 2 image in the MOS1.

```
evselect table=mos1_filt_time.fits withimageset=yes imageset=mos1-b2.fits
   imagebinning=binSize xcolumn=X ximagebinsize=82 ycolumn=Y yimagebinsize=82
   filtertype=expression expression='(FLAG == 0)&&(PI in [500:1000])'
```

and

```
evselect table=mos1_filt_time.fits withimageset=yes imageset=mos1-b5.fits
   imagebinning=binSize xcolumn=X ximagebinsize=82 ycolumn=Y yimagebinsize=82
   filtertype=expression expression='(FLAG == 0)&&(PI in [4500:12000])'
```

The procedure is similar for the MOS2 and PN. Note that, because we are using these images for the specific purpose of source detection, the image binning that we would normally use for MOS (22, corresponding to 1.1 arcsec) must be adjusted to match that of PN (82, corresponding to 4.1 arcsec). For the band 5 images, set the "Selection Expression" text to `(FLAG == 0)&&(PI in [4500:12000])`. We will assume the output images are named `mos1-b2.fits, mos2-b2.fits, pn-b2.fits, mos1-b5.fits, mos2-b5.fits`, and `pn-b5.fits`.

Now we can run *edetect_chain*:

```
edetect_chain
   imagesets='mos1-b2.fits mos1-b5.fits mos2-b2.fits mos2-b5.fits pn-b2.fits pn-b5.fits'
   eventsets='mos1_filt_time.fits mos2_filt_time.fits pn_filt_time.fits'
   attitudeset=attitude.fits likemin=10 witheexpmap=yes
   pimin='500 4500 500 4500 500 4500' pimax='1000 12000 1000 12000 1000 12000'
   ecf='1.70 0.15 1.71 0.15 7.87 0.58' eboxl_list=eboxlist_l.fits
   eboxm_list=eboxlist_m.fits eml_list=emllist.fits esp_withootset=no
```

where

`imagesets` – list of count images
`eventsets` – list of event files
`attitudeset` – attitude file name
`pimin` – list of minimum PI channels for the bands
`pimax` – list of maximum PI channels for the bands
`likemin` – maximum likelihood threshold
`witheexpmap` – create and use exposure maps
`ecf` – energy conversion factors for the bands

eboxl_list – output file name for the local sliding box source detection list
eboxm_list – output file name for the sliding box source detection in background map mode list
eml_list – output file name for maximum likelihood source detection list
esp_withootset – Flag to use an out-of-time processed PN event file, useful in cases where bright point sources have left streaks in the PN data
esp_ooteventset – The out-of-time processed PN event file

7.2.6 Extract the Source and Background Spectra

Throughout the following, please keep in mind that some parameters are instrument-dependent. The parameter specchannelmax should be set to 11999 for the MOS, or 20479 for the PN. Also, for the PN, the most stringent filters, (FLAG==0)&&(PATTERN<=4), must be included in the expression to get a high-quality spectrum.

For the MOS, the standard filters should be appropriate for many cases, though there are some instances where tightening the selection requirements might be needed. For example, if obtaining the best-possible spectral resolution is critical to your work, and the corresponding loss of counts is not important, only the single pixel events should be selected (PATTERN==0). If your observation is of a bright source, you again might want to select only the single pixel events to mitigate pile up (see §7.2.8 and §7.2.9 for a more detailed discussion).

First, make an image of the filtered file, mos1_filt_time.fits, as described in §7.2.1). Download it and display it with ds9, then click on an object whose spectrum you wish to extract. Adjust the extraction region until you are satisfied with it. For this example, we will choose the source at (26165.75, 22816.25) and set the extraction radius to 400.
To extract the source spectrum, type

```
evselect table='mos1_filt_time.fits' energycolumn='PI' withfilteredset=yes
   filteredset='mos1_filt_time_source.fits' keepfilteroutput=yes filtertype='expression'
   expression='((X,Y) in CIRCLE(26165.75,22816.2,400))&&(FLAG==0)'
   withspectrumset=yes spectrumset='mos1_source_pi.fits' spectralbinsize=5
   withspecranges=yes specchannelmin=0 specchannelmax=11999
```

where

table – the event file
energycolumn – energy column
withfilteredset – make a filtered event file
keepfilteroutput – keep the filtered file
filteredset – name of output file
filtertype – type of filter
expression – expression to filter by
withspectrumset – make a spectrum
spectrumset – name of output spectrum
spectralbinsize – size of bin, in eV
withspecranges – covering a certain spectral range
specchannelmin – minimum of spectral range
specchannelmax – maximum of spectral range

The spectrum, in counts per channel, can be viewed with *fv*.

When extracting the background spectrum, follow the same procedures, but change the extraction area. For example, make an annulus around the source; this can be done using two circles, each defining the inner and outer edges of the annulus, then change the filtering expression (and output file name) as necessary.

To extract the background spectrum, type

```
evselect table='mos1_filt_time.fits' energycolumn='PI' withfilteredset=yes
   filteredset='mos1_filt_time_bkg.fits' keepfilteroutput=yes filtertype='expression'
   expression='((X,Y) in CIRCLE(26165.75,22816.2,1200))&&!((X,Y) in CIRCLE(26165.75,22816.2,400))'
   withspectrumset=yes spectrumset='mos1_bkg_pi.fits' spectralbinsize=5
   withspecranges=yes specchannelmin=0 specchannelmax=11999
```

where the keywords are as described above.

7.2.7 Determine the Spectrum Extraction Areas

The source and background region areas can now be found. This is done with the task *backscale*, which takes into account any bad pixels or chip gaps, and writes the result into the BACKSCAL keyword of the spectrum table.

To find the source and background extraction areas, type

```
backscale spectrumset=mos1_source_pi.fits badpixlocation=mos1_filt_time.fits
```

and

```
backscale spectrumset=mos1_bkg_pi.fits badpixlocation=mos1_filt_time.fits
```

7.2.8 Check for Pile Up

Depending on how bright the source is and what modes the EPIC detectors are in, event pile up may be a problem. Pile up occurs when a source is so bright that incoming X-rays strike two neighboring pixels or the same pixel in the CCD more than once in a read-out cycle. In such cases the energies of the two events are in effect added together to form one event. If this happens sufficiently often, 1) the spectrum will appear to be harder than it actually is, and 2) the count rate will be underestimated, since multiple events will be undercounted. To check whether pile up may be a problem, use the SAS task *epatplot*. (Heavily piled sources will be immediately obvious, as they will have a "hole" in the center.) Note that this procedure requires as input the event files created when the spectrum was made.

The output of *epatplot* is a postscript file, which may be viewed with viewers such as *gv*, containing two graphs describing the distribution of counts as a function of PI channel; see Figure 7.3.

A few words about interpretting the plots are in order. The top is the distribution of counts versus PI channel for each pattern class (single, double, triple, quadruple), and the bottom is the expected pattern distribution (**smooth lines**) plotted over the observed distribution (**histogram**). If the lower plot shows the model distributions for single and double events diverging significantly from the observed distributions, then the source is piled up.

The source used in our Lockman Hole example is too faint to provide reasonable statistics for *epatplot* and is far from being affected by pile up. In contrast, plots from two different observations are shown in Figure 7.3 and 7.4. In Figure 7.3, the source is bright enough to provide statistics (and a good fit) at energies above about 1.5 keV. Figure 7.4 shows a plot of a very bright source which is strongly affected by pileup. Note the severe divergence between the model and the observed pattern distribution.

To check for pile up, type

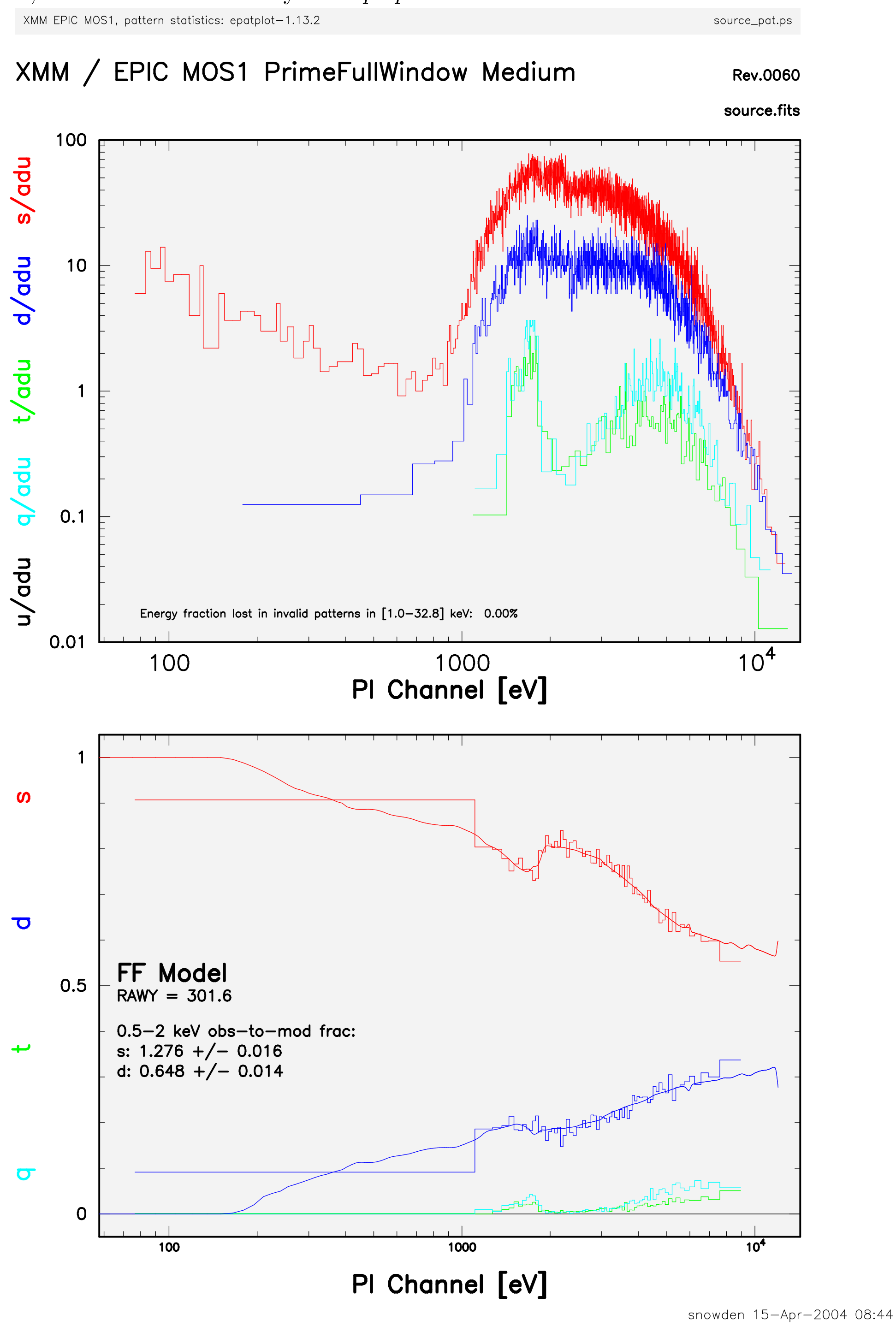
```
epatplot set=mos1_filt_time_source.fits plotfile=mos1_pat.ps
   useplotfile=yes withbackgroundset=yes backgroundset=mos1_filt_time_bkg.fits
```

The postscript file can be copied to the user's local machine and viewed there.

7.2.9 My Observation is Piled Up! Now What?

There are a few ways to deal with pile up. First, using the region selection and event file filtering procedures demonstrated in earlier sections, you can excise the inner-most regions of a source (as they are the most heavily piled up), re-extract the spectrum, and continue your analysis on the excised event file. For this procedure, it is recommended that you take an iterative approach: remove an inner region, extract a spectrum, check with

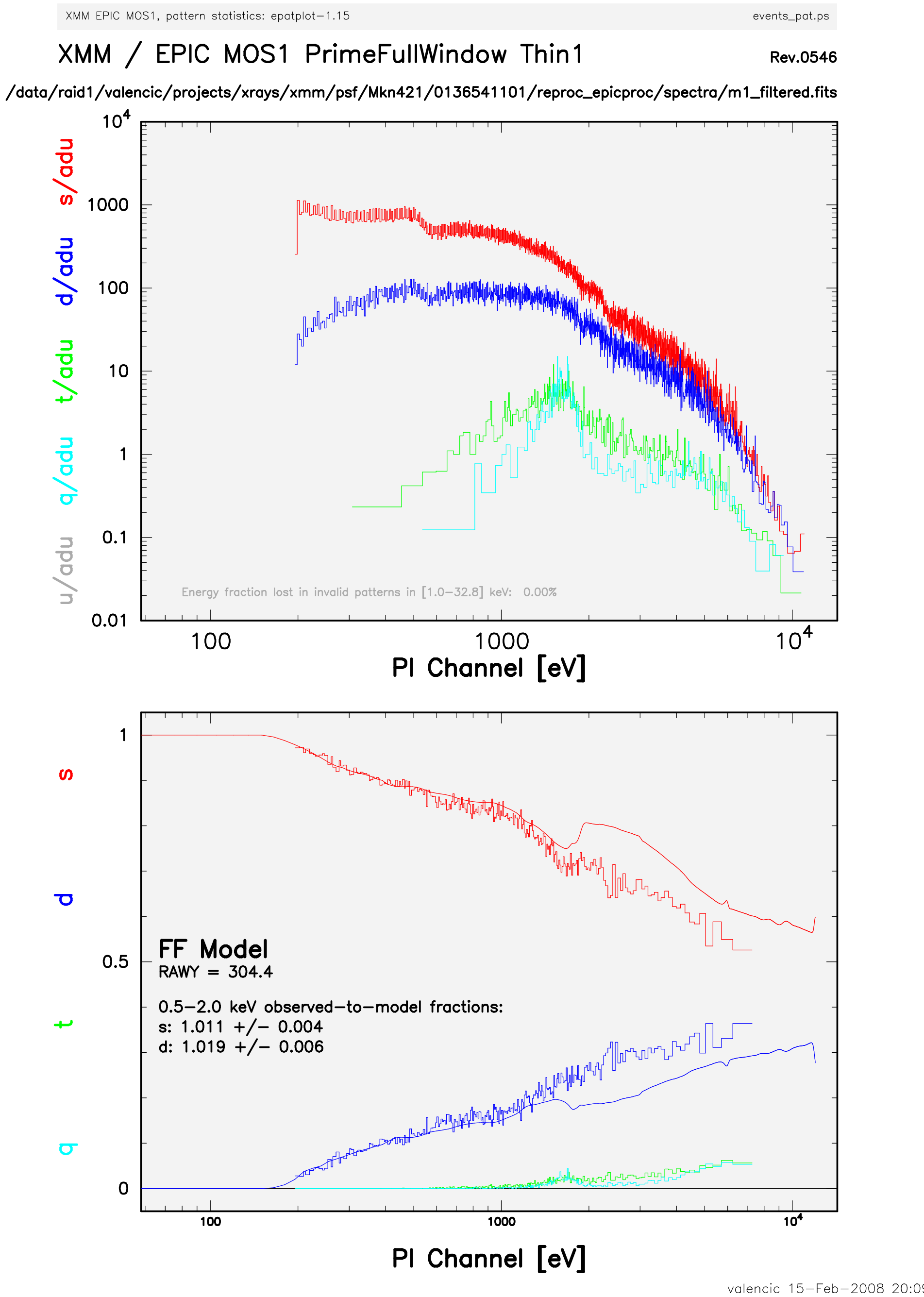
Figure 7.3: The output of *epatplot* for a source without pileup. Note that in the lower plot, for energies less than ~ 1500 eV, there are too few X-rays for *epatplot* to model.



epatplot, and repeat, each time removing a slightly larger region, until the model and observed distribution functions agree.

You can also use the event file filtering procedures to include only single pixel events (`PATTERN==0`), as these events are less sensitive to pile up than other patterns.

Figure 7.4: The output of *epatplot* for a heavily piled source. In the lower plot, there are large differences between the predicted and observed pattern distribution at energies above $\sim$ 1000 eV.



7.2.10 Create the Photon Redistribution Matrix (RMF) and Ancillary File (ARF)

In order to do spectral analysis, it is necessary to find the instrument's repsonse as a function of energy and PI channel. This is done by reformating the detector response and energy bounds information and correcting for instrumental effects, and writing the result to the Redistribution Matrix File (RMF). The following assumes that an appropriate source spectrum, named `mos1_source_pi.fits`, has been extracted as in §7.2.6.

To make the RMF, type

```
rmfgen rmfset=mos1_rmf.fits spectrumset=mos1_source_pi.fits
```

where

`rmfset` – output file
`spectrumset` – spectrum file

Now use the RMF, spectrum, and event file to make the ancillary file.
To make the ARF, type

```
arfgen arfset=mos1_arf.fits spectrumset=mos1_source_pi.fits withrmfset=yes
   rmfset=mos1_rmf.fits withbadpixcorr=yes badpixlocation=mos1_filt_time.fits
```

where

`arfset` – output ARF file name
`spectrumset` – input spectrum file name
`withrmfset` – flag to use the RMF
`rmfset` – RMF file created by *rmfgen*
`withbadpixcorr` – flag to include the bad pixel correction
`badpixlocation` – file containing the bad pixel information; should be set to the event file from which the spectrum was extracted.

At this point, the spectrum is ready to be analyzed, so we can prepare the spectrum for fitting.

7.2.11 Prepare and Fit the Spectrum

With the source and background spectra now extracted and the RMF and ARF created, we will do some simple spectral fitting. SAS does not include fitting software, so HEASoft packages will be used, and all fitting tasks will be called from the Command Window.
Nearly all spectra will need to be binned for statistical purposes. The procedure *grppha*, located in the HEASARC folder in the Available Tools window, provides an excellent mechanism to do just that.
The following commands not only group the source spectrum for Xspec but also associate the appropriate background and response files for the source.

1) In the Command Window, type:

   ```
   grppha
   ```

   and edit the parameters and file names as appropriate:

   ```
   Please enter PHA filename[] mos1_source_pi.fits ! input spectrum file name
   Please enter output filename[] mos1_grp.fits    ! output grouped spectrum
   GRPPHA[] chkey BACKFILE mos1_bkg_pi.fits        ! include the background spectrum
   GRPPHA[] chkey RESPFILE mos1_rmf.fits           ! include the RMF
   GRPPHA[] chkey ANCRFILE mos1_arf.fits           ! include the ARF
   GRPPHA[] group min 25                           ! group the data by 25 counts/bin
   GRPPHA[] exit
   ```

Upon exiting, the output file `mos1_grp.fits` will appear in your working directory.
Next, use Xspec to fit the spectrum by typing:

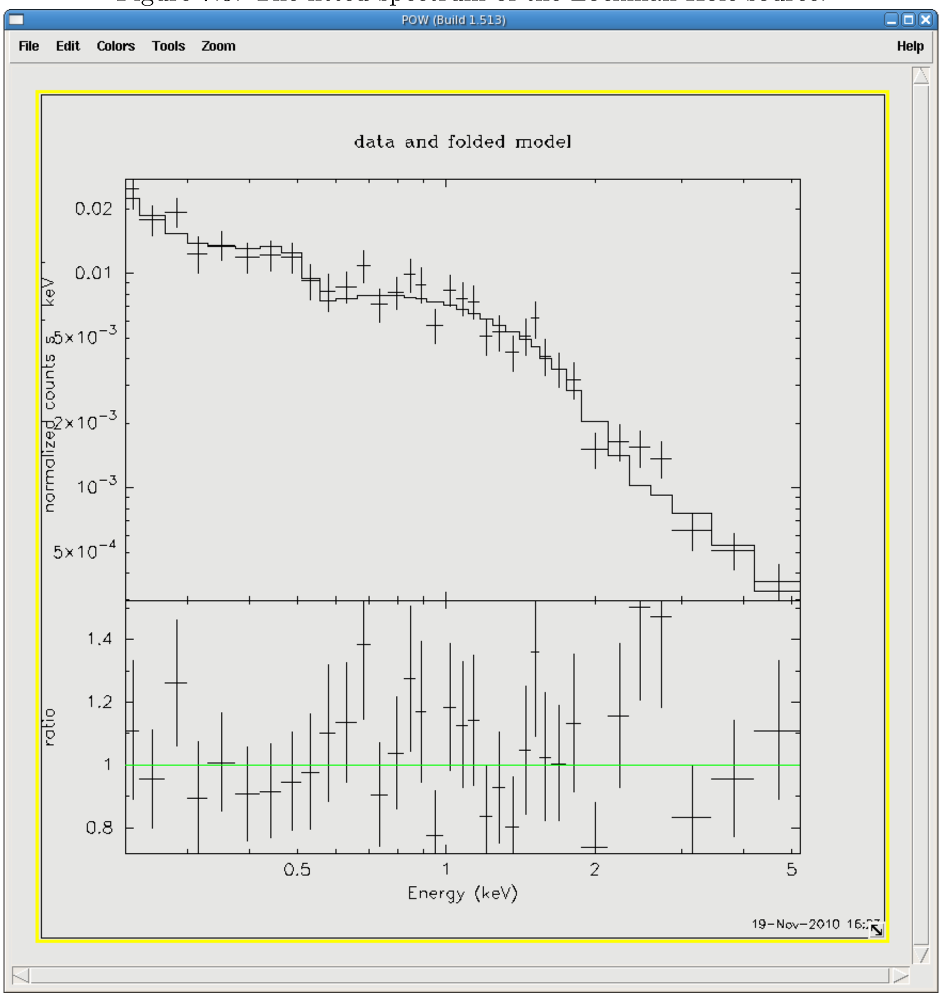
```
xspec
```

A POW window will pop up and display the spectrum later on. Edit the parameters and file names as appropriate:

```
XSPEC> data mos1_grp.fits         ! input data
XSPEC> ignore 0.0-0.2,6.6-**      ! ignore unusable energy ranges, in keV
                                  ! set a range appropriate for the data
XSPEC> model wabs(pow+pow)        ! set spectral model to two absorbed power laws
1:wabs:nH> 0.01                   ! set model absorption column density to 1.e20
2:powerlaw:PhoIndex> 2.0          ! set the first model power law index to -2.0
3:powerlaw:norm>                  ! use the default model normalization
4:powerlaw:PhoIndex> 1.0          ! set the second model power law index to -1.0
5:powerlaw:norm>                  ! use the default model normalization
renorm                            ! renormalize the model spectrum
XSPEC> fit                        ! fit the model to the data
XSPEC> setplot energy             ! plot energy along the X axis
XSPEC> plot ldata ratio           ! plot two panels with the log of the data and
                                  !     the data/model ratio values along the Y axes
XSPEC> exit                       ! exit Xspec
```

Figure 7.5 shows the fit to the spectrum.

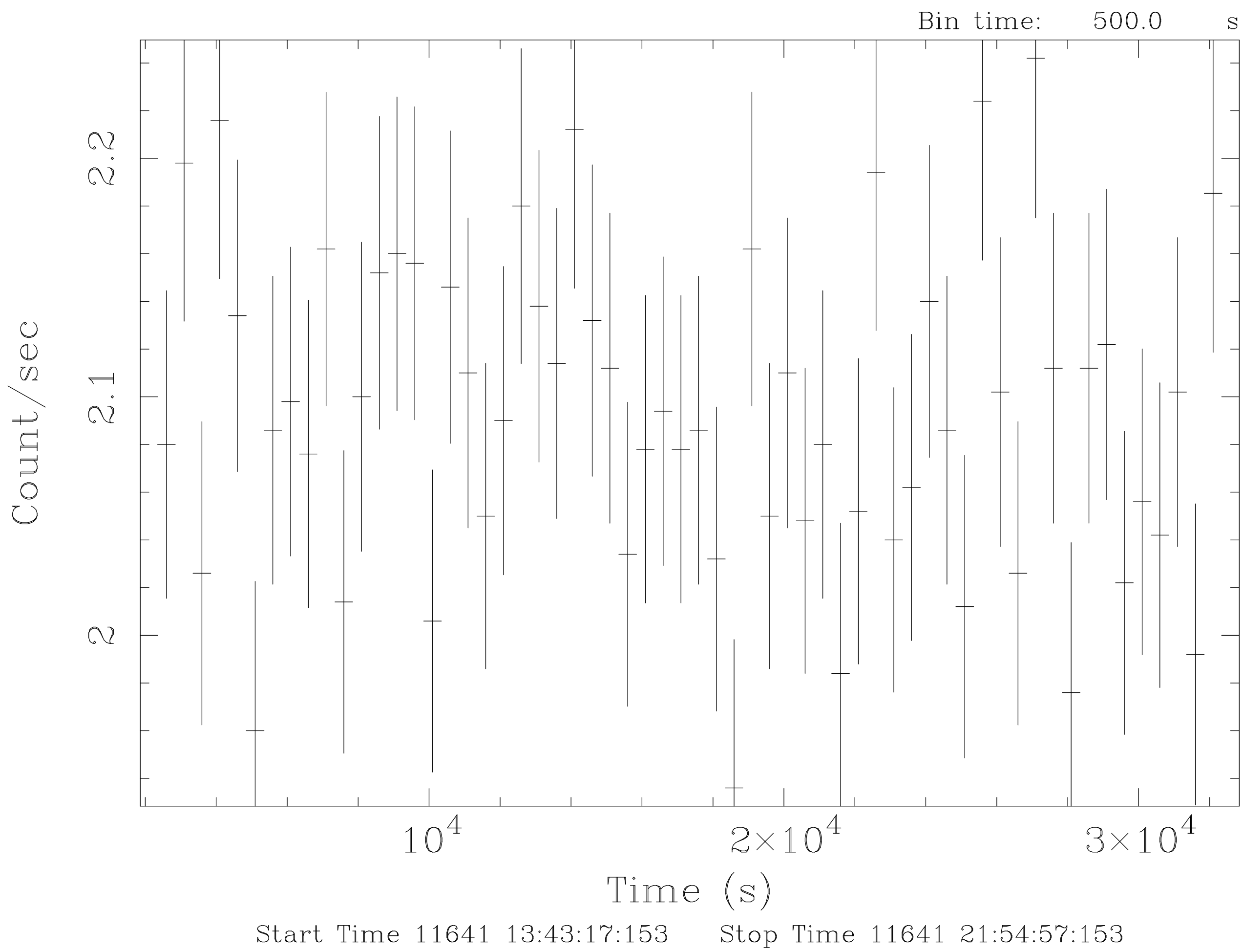
Figure 7.5: The fitted spectrum of the Lockman Hole source.



7.3 Timing Analysis

This section will demonstrate some basic timing analysis of EPIC image-mode data using the Xronos analysis package. For this exercise, the central source from the observation of G21.5-09 (Obs ID 0122700101) is used. These examples assume that the source's lightcurve has been made as in §7.2.2, but with `timebinsize` set to `1` and `makeratecolumn` set to `no`; the name of this file assumed to be `source_ltcrv.fits`.
For the aficionado, the task *barycen* can be used for the barycentric correction of the source event arrival times. The Xronos tools can be access through Hera by clicking on XRONOS in the Available Tools panel.

Figure 7.6: Light curve for the source analyzed as in §7.2.2.



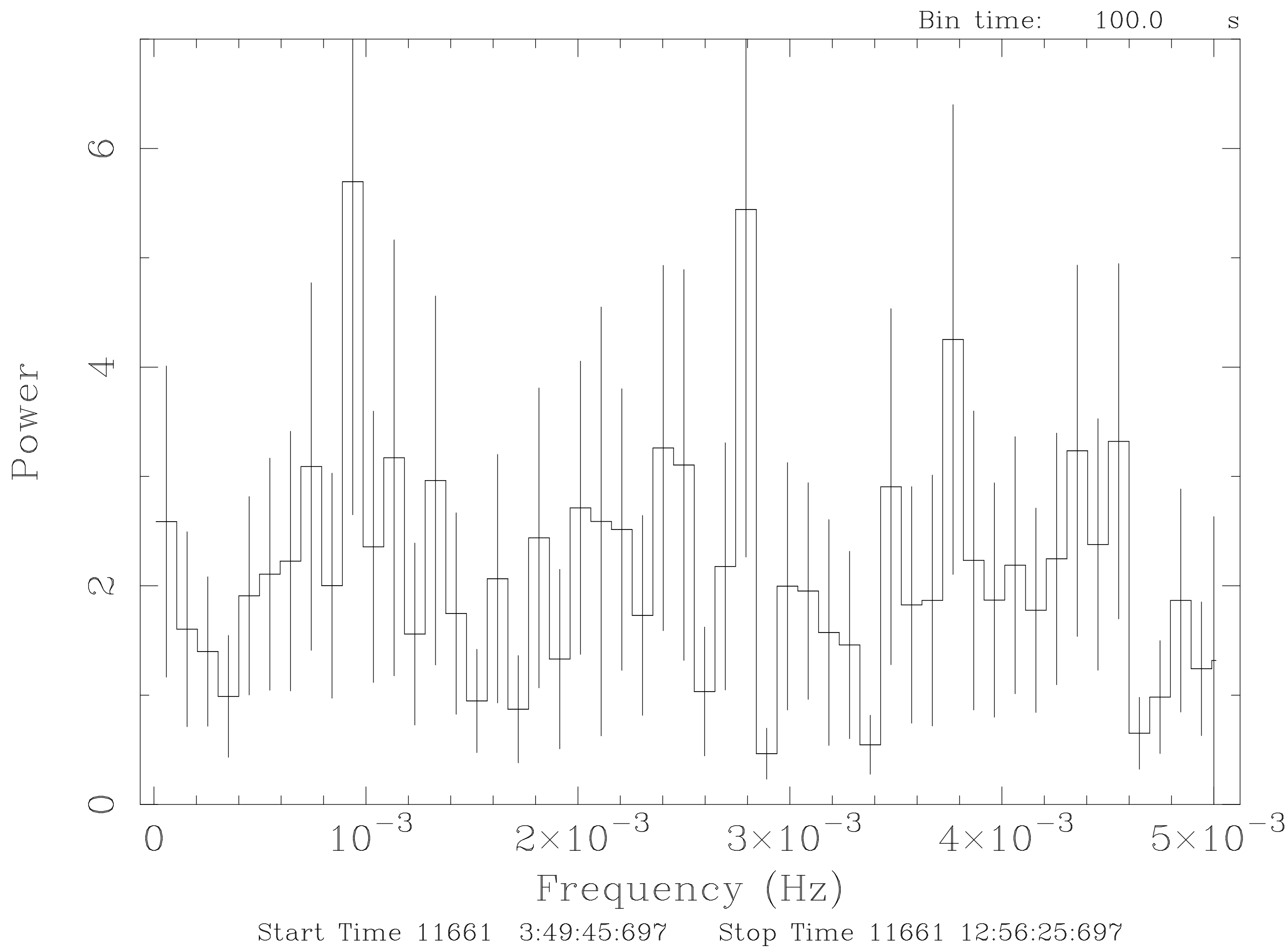
To make a binned lightcurve, type:

```
lcurve nser=1 cfile1='source_ltcrv.fits' window=- dtnb=500 nbint=450
   outfile='lightcurve_binned.fits' plot=no
```

where

`nser` – number of time series
`cfile1` – filename first series
`window` – name of window file (if a subset of the time series is required)
`dtnb` – bin size (time)
`nbint` – number of bins per interval
`outfile` – output file name (FITS format light curve)
`plot` – plot flag

Figure 7.7: Power spectrum density for the source.



The output can be viewed with *fv* by right-clicking on the filename and selecting the "Edit/Display File" option. Output is shown in Figure 7.6.

To calculate power spectrum density, type

```
powspec cfile1='source_ltcrv.fits' window=- dtnb=100.0 nbint=300
   nintfm=INDEF rebin=5 plot=no outfile='power.fits'
```

where

`cfile1` – filename first series
`window` – name of window file (if a subset of the time series is required)
`dtnb` – bin size (time)
`nbint` – number of bins per interval
`nintfm` – number of intervals in each power spectrum
`rebin` – rebin factor for power spectrum (0 for no rebinning)
`plot` – plot flag
`outfile` – output file name (FITS format power spectrum)

The output can be viewed with *fv* by right-clicking on the filename and selecting the "Edit/Display File" option. Output is shown in Figure 7.7.

To search for periodicities in the time series, type:

```
efsearch cfile1=source_ltcrv.fits window=- sepoch=INDEF dper=20 nphase=10
   nbint=INDEF nper=100 dres=INDEF plot=no outfile=autocor.fits
```

where

`cfile1` – filename first series
`window` – name of window file (if a subset of the time series is required)
`sepoch` – value for epoch used for phase zero when folding the time series
`dper` – value for the period used in the folding
`nphase` – number of phases per period
`nbint` – number of bins per interval
`nper` – number of sampled periods during search
`dres` – sampling resolution of search
`plot` – plot flag
`outfile` – output file name (FITS format)

To calculate the autocorrelation for a time series, type:

```
autocor cfile1=source_ltcrv.fits window=- dtnb=24.0 nbint=2048 nintfm=INDEF
   rebin=0 plot=no outfile=auto.fits
```

where

`cfile1` – filename first series
`window` – name of window file (if a subset of the time series is required)
`dtnb` – bin size (time)
`nbint` – number of bins per interval
`nintfm` – number of intervals to be summed in each autocorrelation function
`rebin` – rebin factor for autocorrelation function (0 for no rebinning)
`plot` – plot flag
`outfile` – output file name (FITS format autocorrelation spectrum)

To calculate statistical quantities for a time series, type

```
lcstats cfile1=source_ltcrv.fits window=- dtnb=6.0 nbint=8192
```

where

`cfile1` – filename first series
`window` – name of window file
`dtnb` – integration time (binning)
`nbint` – number of bins

The output will be written in the Command Window.

Chapter 8

An RGS Data Processing and Analysis Primer

While a variety of analysis packages can be used for the following steps, the SAS was designed for the basic reduction and analysis of XMM-Newton data and will therefore be used here for demonstration purposes.

At this point, it is assumed that you have downloaded the data from the HEASARC archive onto a Hera server, standard or anonymous Hera is running (see §4.2), you have prepared the data for processing with *odfingest* (see §6), and the working directory PROC has been made. Throughout this chapter, we will use the Mkn 421 dataset with ObsID 0153950701 available through links at the HEASARC archive.

8.1 Rerunning the Pipeline

It is very likely that you will want to filter your data to some extent; in this case, you will need to reprocess it in order to determine the appropriate filters, regardless of the age of the observation. To do this, verify that the working directory PROC is highlighted in the GUI. In the new Command Window you made at the end of §6, run the task(s):

```
rgsproc orders='1 2' bkgcorrect=no withmlambdacolumn=yes spectrumbinning=lambda
```

where

`orders` – dispersion orders to extract
`bkgcorrect` – subtract background from source spectra?
`withmlambdacolumn` – include a wavelength column in the event file product
`spectrumbinning` – accumulate the spectrum either in wavelength or beta space

This takes several minutes, and outputs 12 files per RGS, plus 3 general use FITS files. At this point, renaming files to something easy to type is a good idea. This is easily done by right-clicking on the event files. We will assume that the newly pipelined event files are named `rgs1.fits` and `rgs2.fits`.

8.2 Potentially useful tips for using the pipeline

The pipeline task, *rgsproc*, is very flexible and can address potential pitfalls for RGS users. In §8.1, we used the default parameter settings, and if this is sufficient for your data (and it should be for most), feel free to skip to §8.3. In the following sections, we will look at the cases of a nearby bright optical source, a nearby bright X-ray source, and a user-defined source.

8.2.1 A Nearby Bright Optical Source

With certain pointing angles, zeroth-order optical light may be reflected off the telescope optics and cast onto the RGS CCD detectors. If this falls on an extraction region, the current energy calibration will require a wavelength-dependent zero-offset. Stray light can be detected on RGS DIAGNOSTIC images taken before, during and after the observation. This test, and the offset correction, are not performed on the data before delivery. To check for stray light and apply the appropriate offsets, enter

```
rgsproc orders='1 2' bkgcorrect=no calcoffsets=yes withoffsethistogram=no
```

where the parameters are as described in §8.1 and

`calcoffsets` – calculate PHA offsets from diagnostic images
`withoffsethistogram` – produce a histogram of uncalibrated excess for the user

8.2.2 A Nearby Bright X-ray Source

In the example above, it is assumed that the field around the source contains sky only. Provided a bright background source is well-separated from the target in the cross-dispersion direction, a mask can be created that excludes it from the background region. Here the source has been identified in the EPIC images and its coordinates have been taken from the EPIC source list which is included among the pipeline products. The bright neighboring object is found to be the third source listed in the sources file. The first source is the target:

```
rgsproc orders='1 2' bkgcorrect=no withepicset=yes
   epicset=PiiiiiijjkkaablllEMSRLInmmm.FTZ exclsrcsexpr='INDEX==1&&INDEX==3'
```

where the parameters are as described in §8.1 and

`withepicset` – calculate extraction regions for the sources contained in an EPIC source list
`epicset` – name of the EPIC source list, such as generated by *emldetect* or *eboxdetect* procedures
`exclsrcsexpr` – expression to identify which source(s) should be excluded from the background extraction region

8.2.3 User-defined Source Coordinates

If the true coordinates of an object are not included in the EPIC source list or the science proposal, the user can define the coordinates of a new source by entering:

```
rgsproc orders='1 2' bkgcorrect=no withsrc=yes srclabel=Mkn421 srcstyle=radec
   srcra=166.113808 srcdec=+38.208833
```

where the parameters are as described in §8.1 and

`withsrc` – make the source be user-defined
`srclabel` – source name
`srcstyle` – coordinate system in which the source position is defined
`srcra` – the source's right ascension in decimal degrees
`srcdec` – the source's declination in decimal degrees

8.3 Examine and Analyze the Data

Since the event files are current, we can proceed with some simple analysis demonstrations, which will allow us to generate filters. Note that the Command Window always sees (and places) files in the directory that it was invoked in.

8.3.1 Create and Display an Image

Two commonly-made plots are those showing `PI` vs. `BETA_CORR` (also known as "banana plots") and `XDSP_CORR` vs. `BETA_CORR`.
To create such images, type

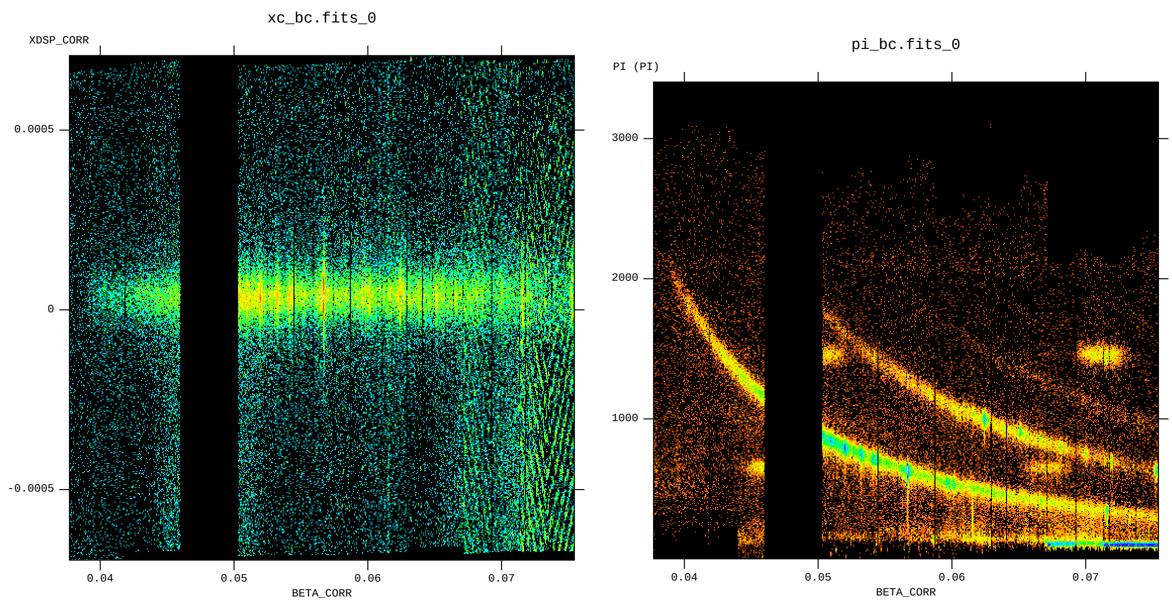
```
evselect table=rgs1.fits withimageset=yes
   imageset=pi_bc.fits xcolumn=BETA_CORR ycolumn=PI
   imagebinning=imageSize ximagesize=600 yimagesize=600
```

where

> `table` – input event table
>
> `withimageset` – make an image
>
> `imageset` – name of output image
>
> `xcolumn` – event column for X axis
>
> `ycolumn` – event column for Y axis
>
> `imagebinning` – form of binning, force entire image into a given size or bin by a specified number of pixels
>
> `ximagesize` – output image pixels in X
>
> `yimagesize` – output image pixels in Y

Plots comparing `BETA_CORR` to `XDSP_CORR` may be made in a similar way. The output files can be viewed by using a standard FITS display. The example plots, as seen with *fv*, are shown in Figure 8.1.

Figure 8.1: Plots of XDSP_CORR vs. BETA_CORR (left) and PI vs. BETA_CORR (right). The gap is due to the missing CCD7. Similarly, CCD4 is missing in RGS2.



8.3.2 Create and Display a Light Curve

The background is assessed through examination of the light curve. We will extract a region, CCD9, that is most susceptible to proton events and generally records the least source events due to its location close to the optical axis. Also, to avoid confusing solar flares for source variability, a region filter that that removes the source from the final event list should be used. The region filters are kept in the source file product `P*SRCLI_*.FIT`.

More experienced users should be aware that with SAS 13, the `*SRCLI*` file's **column information changed**. *rgsproc* now outputs an M_LAMBDA column instead of BETA_CORR, and M_LAMBDA should be used to generate the light curve. (The `*SRCLI*` file that came with the PPS products still contains a BETA_CORR column if you prefer to use that instead.)

To create a light curve, type

```
evselect table=rgs1.fits withrateset=yes rateset=r1_ltcrv.fits
   maketimecolumn=yes timebinsize=100 makeratecolumn=yes
   expression=
   '(CCDNR==9)&&(REGION(P0153950701R1S001SRCLI_0000.FIT:RGS1_BACKGROUND,M_LAMBDA,XDSP_CORR))'
```

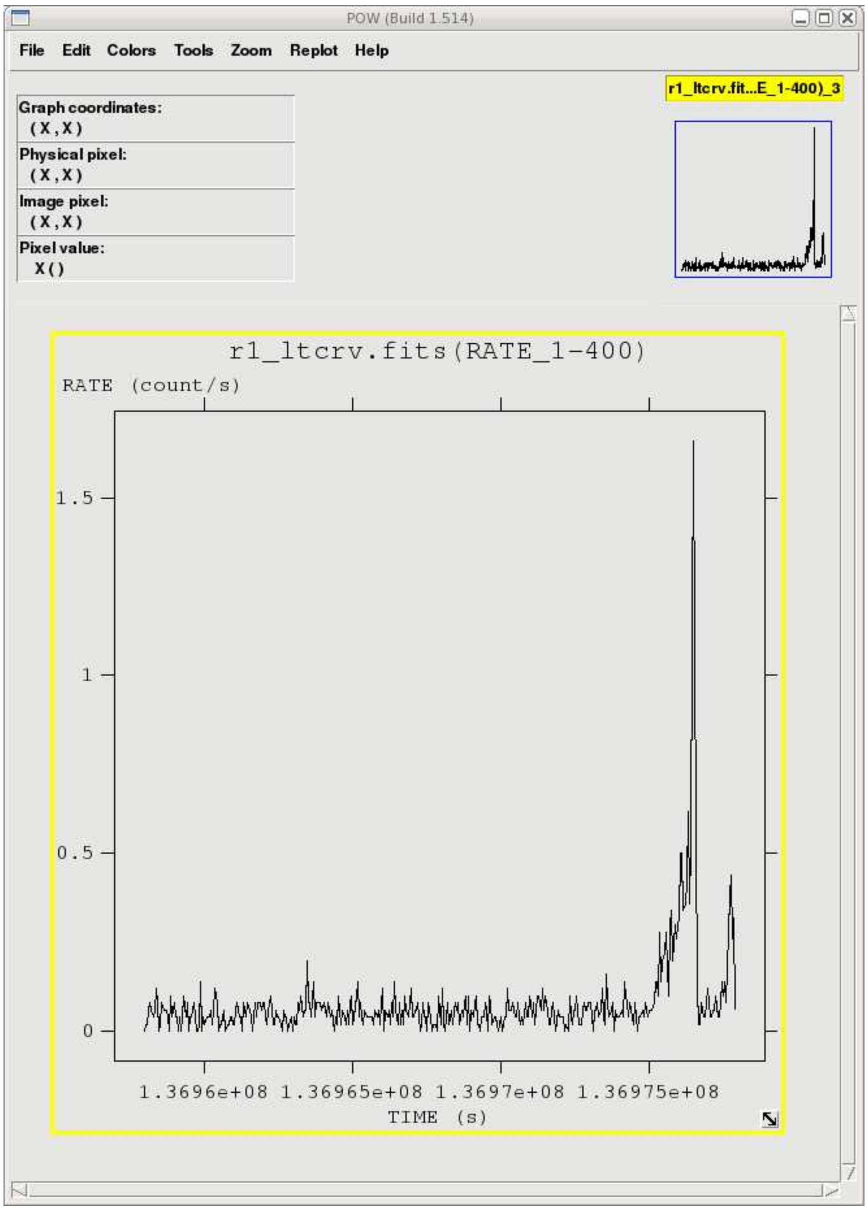
where

`table` – input event table
`withrateset` – make a light curve
`rateset` – name of output light curve file
`maketimecolumn` – control to create a time column
`timebinsize` – time binning (seconds)
`makeratecolumn` – control to create a count rate column, otherwise a count column will be created

`expression` – filtering criteria

The output file `r1_ltcrv.fits` can be viewed with *fv*. The light curve is shown in Figure 8.2.

Figure 8.2: Background event rate from the RGS1 CCD9 chip. The flares are solar events. The time units are elapsed mission time.



8.3.3 Generating the Good Time Interval (GTI) File

Examination of the lightcurve shows that there is a noisy section at the end of the observation, after 1.36975e8 seconds, where the count rate is well above the normal background count rate of ~ 0.05 count/second. There are two procedures that make the GTI file (*gtibuild* and *tabgtigen*) that, when applied to the event file in another run of *rgsproc*, will excise these sections.

The first method, using *gtibuild*, requires a text file as input. This file can be made on your local machine and uploaded to your Hera account by right-clicking and dragging the file from your local directory to the remote directory. In the first two columns, refer to the start and end times (in seconds) that you are interested in, and in the third column, indicate with either a + or - sign whether that region should be kept or removed. In the example case, then, we would write in our ASCII file (named `r1_gti.txt`):

```
1.36958e8 1.36975e8 +
```

and proceed to the task *gtibuild*:

```
gtibuild file=r1_gti.txt table=r1_gti.fits
```

where

`file` – intput text file
`table` – output gti table

Alternatively, we can make the GTI file with *tabgtigen* and filter for RATE (though we could just as easily filter on TIME) by entering

```
tabgtigen table=r1_ltcrv.fits gtiset=r1_gti.fits expression='RATE<0.2'
```

where

`table` – the lightcurve file
`gtiset` – output gti table
`expression` – the filtering criteria. Since the nominal count rate is 0.05 about count/sec, we have set the upper limit to 0.2 count/sec.

8.3.4 Applying the GTI

Now that we have GTI file, we can apply it to the event file by running *rgsproc* again. *rgsproc* is a complex task, running several steps, with five different entry and exit points. It is not necessary to rerun all the steps in the procedure, only the ones involving filtering.

To apply the GTI to the event file, type

```
rgsproc orders='1 2' auxgtitables=r1_gti.fits bkgcorrect=no
   withmlambdacolumn=yes entrystage=3:filter finalstage=5:fluxing
```

where

`orders` – spectral orders to be processed
`auxgtitables` – gti file in FITS format
`bkgcorrect` – subtract background from source spectra?
`withmlambdacolumn` – include a wavelength column in the event file product
`entrystage` – stage at which to begin processing
`finalstage` – stage at which to end processing

We will refer to the output event file as `r1_filt.fits`.

8.3.5 Creating the Response Matrices (RMFs)

Response matrices (RMFs) are not provided as part of the pipeline product package, so you must create your own before analyzing data. The task *rgsproc* generates a response matrix automatically, but as noted in §8.2.3, the source coordinates are under the observer's control. The source coordinates have a profound influence on the accuracy of the wavelength scale as recorded in the RMF that is produced automatically by *rgsproc*, and each RGS instrument and each order will have its own RMF.

Making the RMF is easily done with the package *rgsrmfgen*. Please note that, unlike with EPIC data, it is not necessary to make ancillary response files (ARFs).

To make the RMFs, type

```
rgsrmfgen spectrumset=P0153950701R1S001SRSPEC1001.FIT rmfset=r1_o1_rmf.fits
   evlist=r1_filt.fits emin=0.4 emax=2.5 rows=5000
```

where

`spectrumset` – spectrum file
`evlist` – event file
`emin` – lower energy limit of the response file
`emax` – upper energy limit of the response file
`rows` – number of energy bins; this should be greater than 3000
`rmfset` – output FITS file

At this point, the spectra can be analyzed. If you you wish, skip the discussion on combining spectra (§8.3.6) and go straight to fitting the spectrum (§8.4.)

8.3.6 Combining Spectra

Spectra from the same order in RGS1 and RGS2 can be safely combined to create a spectrum with higher signal-to-noise if they were reprocessed using *rgsproc* with `spectrumbinning=lambda`, as we did in §8.1 (this also happens to be the default). The task *rgscombine* also merges response files and background spectra. When merging response files, be sure that they have the same number of bins. For this example, we assume that RMFs were made for order 1 in both RGS1 and RGS2.

To merge RGS1 and RGS2 spectra, type

```
rgscombine pha='P0153950701R1S001SRSPEC1001.FIT P0153950701R2S002SRSPEC1001.FIT'
   rmf='r1_o1_rmf.fits r2_o1_rmf.fits'
   bkg='P0153950701R1S001BGSPEC1001.FIT P0153950701R2S002BGSPEC1001.FIT'
   filepha='r12_o1_srspec.fits' filermf='r12_o1_rmf.fits'
   filebkg='r12_o1_bgspec.fits' rmfgrid=5000
```

where

`pha` – list of spectrum files
`rmf` – list of response matrices
`bkg` – list of bakcground spectrum files
`filepha` – output merged spectrum
`filermf` – output merged response matrix
`filebkg` – output merged badkground spectrum
`rmfgrid` – number of energy bins; should be the same as the input RMFs

The spectra are ready for analysis, so we can prepare the spectrum for fitting.

8.4 Approaches to Spectral Fitting and the Cash Statistic

For data sets of high signal-to-noise and low background, where counting statistics are within the Gaussian regime, the data products above are suitable for analysis using the default fitting scheme in XSPEC, χ^2-minimization. However, for low count rates, in the Poisson regime, χ^2-minimization is no longer suitable. With low count rates in individual channels, the error per channel can dominate over the count rate. Since channels are weighted by the inverse-square of the errors during χ^2 model fitting, channels with the lowest count rates are given overly-large weights in the Poisson regime. Spectral continua are consequently often fit incorrectly, with the model lying underneath the true continuum level. This will be a common problem with most RGS sources. Even if count rates are large, much of the flux from these sources can be contained within emission lines, rather than the continuum. Consequently, even obtaining correct equivalent widths for such sources is non-trivial.

The traditional way to increase the signal-to-noise of a data set is to rebin or group the channels, since, if channels are grouped in sufficiently large numbers, the combined signal-to-noise of the groups will jump into the Gaussian regime. However, this results in the loss of information. For example, sharp features like an absorption edge or emission line can be completely washed out. Further, in the Poisson regime, the background spectrum cannot simply be subtracted, as is commonly done in the Gaussian regime, since this could result in negative counts. Therefore, rebinning should be reserved for fast, preliminary analysis of spectra without sharp features, or for making plots for publication. When working on the final analysis for a low-count data set, the (unbinned) background and source spectra should be fitted simultaneously using the Cash statistic. (If fitting with XSPEC, be sure you are running v11.1.0 or later. This is because RGS spectrum files have prompted a slight modification to the OGIP standard, since the RGS spatial extraction mask has a spatial-width which is a varying function of wavelength. Thus, it has become necessary to characterize the `BACKSCL` and `AREASCL` parameters as vectors (i.e., one number for each wavelength channel), rather than scalar keywords as they are for data from the EPIC cameras and past missions. These quantities map the size of the source extraction region to the size of the background extraction region and are essential for accurate fits. Only Xspec v11.1.0, or later versions, are capable of reading these vectors, so be certain that you have an up-to-date installation at your site.)

Finally, a caveat of using the Cash statistic in Xspec is that the scheme requires a "total" and "background" spectrum to be loaded into Xspec. This is in order to calculate parameter errors correctly. Consequently, be sure not to use the "net" spectra that were created as part of product packages by SAS v5.2 or earlier. To change schemes in Xspec before fitting the data, type:

```
XSPEC> statistic cstat
```

For our sample spectrum, we will rebin and fit it with χ^2 statistics.

8.4.1 Spectral Rebinning

There are two ways to rebin a spectrum: the FTOOL *grppha*, or the RGS pipeline. *grppha* can group channels using an algorithm which bins up consecutive channels until a count rate threshold is reached. This method conserves the resolution in emission lines above the threshold while improving statistics in the continuum.

To rebin the spectrum with *grppha*, type

```
grppha
```

and edit the parameters as needed:

```
> Please enter PHA filename[] P0153950701R1S001SRSPEC1001.FIT
> Please enter output filename[] P0153950701R1S001SRSPEC1001.bin30.FIT
> GRPPHA[] group min 30
> GRPPHA[] exit
```

The disadvantage of using *grppha* is that, although channel errors are propagated through the binning process correctly, the errors column in the original spectrum product is not strictly accurate. The problem arises because there is no good way to treat the errors within channels containing no counts. To allow statistical fitting, these channels are arbitrarily given an error value of unity, which is subsequently propagated through the binning. Consequently, the errors are overestimated in the resulting spectra.

The other approach, which involves calling the RGS pipeline after it is complete, bins the data during spectral extraction. The following rebins the pipeline spectrum by a factor 3.
To rebin the spectrum with *rgsproc*, type

```
rgsproc orders='1 2' rebin=3 rmfbins=5000 entrystage=4:spectra
   finalstage=5:fluxing bkgcorrect=no
```

where

`orders` – dispersion orders to extract
`rebin` – wavelength rebinning factor
`rmfbins` – number of bins in the response file; this should be greater than 3000
`entrystage` – entry stage to the pipeline
`finalstage` – exit stage for the pipeline

One disadvantage of this approach is that you can only choose integer binning of the original channel size. To change the sampling of the events, the pipeline must be run from the second stage ("angles") or earlier:

```
rgsproc orders='1 2' nbetabins=1133 rmfbins=5000 entrystage=2:angles
   finalstage=fluxing bkgcorrect=no
```

where the parameters are as defined previously, and

`nbetabins` – number of bins in the dispersion direction; the default is 3400

The disadvantage of using *rgsproc*, as opposed to *grppha*, is that the binning is linear across the dispersion direction. Velocity resolution is lost in the lines, so the accuracy of redshift determinations will be degraded, transition edges will be smoothed, and neighboring lines will become blended.

8.5 Fitting a Spectral Model

We can fit the spectrum using Xspec. This is easily done by entering

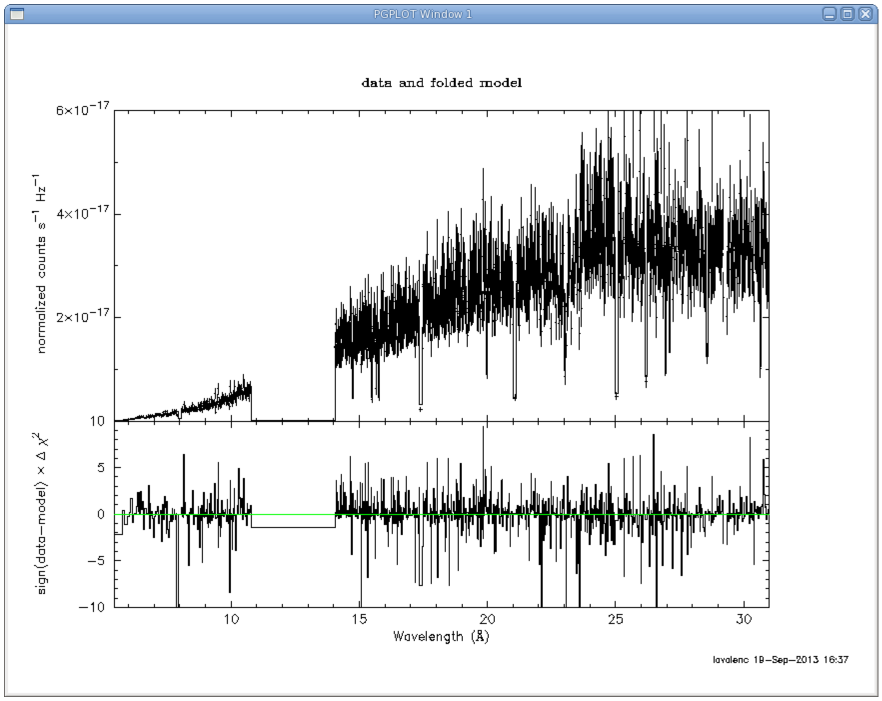
```
xspec
```

Enter the data, background, and response file at the prompts, and edit the fitting parameters as needed.

```
XSPEC> data P0153950701R1S001SRSPEC1001.bin30.FIT ! input data
XSPEC> back P0153950701R1S001BGSPEC1001.bin30.FIT ! input background
XSPEC> resp r1_o1_rmf.fits                        ! input response file
XSPEC> ignore **-0.4                              ! set sensible limits
XSPEC> model wabs*pow                             ! set spectral model to absorbed powerlaw
1:wabs:nH> 0.01                                   ! enter reasonable initial values
2:powerlaw:PhoIndex>2.0
3:powerlaw:norm>1.0
XSPEC> renorm
XSPEC> fit
XSPEC> cpd /xw
XSPEC> setplot wave
XSPEC> setplot command window all
XSPEC> setplot command log x off
XSPEC> setplot command wind 1
XSPEC> setplot command r y 0 6e-17
XSPEC> setplot command wind 2
XSPEC> setplot command r y -10 10
XSPEC> plot data chi
XSPEC> exit
```

Figure 8.3 shows the fit to the spectrum.

Figure 8.3: 1st order RGS1 spectrum of Mkn 421. The fit is an absorbed power law model. The gap between 10–15Å is due to the absence of CCD7.



8.6 Analysis of Extended Sources

8.6.1 Region masks

The optics of the RGS allow spectroscopy of reasonably extended sources, up to a few arc minutes. The width of the spatial extraction mask is defined by the fraction of total events one wishes to extract. With the default pipeline parameter values, over 90% of events are extracted, assuming a point-like source.

Altering and optimizing the mask width for a spatially-extended source may take some trial and error, and, depending on the temperature distribution of the source, may depend on which lines one is currently interested in. While Mkn 421 is not an extended source, the following example increases the width of the extraction mask and ensures that the size of the background mask is reduced so that the two do not overlap.

To adjust the region mask with *rgsproc*, enter

```
rgsproc orders='1 2' entrystage=4:spectra finalstage=5:fluxing bkgcorrect=no
   xpsfincl=99 xpsfexcl=99 pdistincl=95
```

where parameters are as they were decsribed previously, and

`xpsfincl` – include this fraction of point-source events inside the spatial source extraction mask
`xpsfexcl` – exclude this fraction of point-source events from the spatial background extraction mask
`pdistincl` – include this fraction of point-source events inside the pulse height extraction mask

Observing extended sources effectively broadens the psf of the spectrum in the dispersion direction. Therefore, it is prudent to also increase the width of the PI masks using the `pdistincl` parameter in order to prevent event losses.

8.6.2 Making RMFs for extended sources

RGS response matrices as made in §8.3.5 are appropriate for use with point sources only. If we are interested in analyzing an extended source, the RMF must take into account the spatial degradation of the resolution.

The most straight-forward way to do this is to modify the response matrix prior to spectral fitting. For sources extended up to about 1 arcminute, this can be done with the FTOOL *rgsrmfsmooth*. It requires three files: the point source RMF (as made in §8.3.5), an image of the source (from an EPIC camera, see §7.2.1, or different mission), and a text file. The better the resolution of the image, the more accurate the modified RMF will be, so if a Chandra image is available for a source, it should be used instead of an EPIC image. The text file must list the name of the image, the boresight, and the aperture size in the following format:

```
RGS_XSOURCE_IMAGE <name of source image>
RGS_XSOURCE_BORESIGHT <image boresight:  RA (h:m:s), DEC(degrees:m:s), and PA (decimal
   degrees)>
RGS_XSOURCE_EXTRACTION <source radius in arcminutes>
```

For an example case, we will name our text file `xsource.mod`. We will assume that a RMF for the first order grating was made as in §8.3.5 and an MOS1 image was made as in §7.2.1; `xsource.mod` contains these lines:

```
RGS_XSOURCE_IMAGE my_MOS1_image.fits
RGS_XSOURCE_BORESIGHT 05:28:45 -65:26:55 219.769546508789
RGS_XSOURCE_EXTRACTION 1.2
```

This file can be made on the user's local machine and uploaded to the Hera server by right-clicking and dragging the file from the Local Directory panel to the Remote Directory. Then, type

```
rgsrmfsmooth rmffil=my_ps_rmf.fits imgfil=xsource.mod order=1 outfil=my_es_rmf.fits
```

where

`rmffil` – the point source RGS RMF
`imgfil` – the text file with the name of the source image, boresight, and extraction region
`order` – grating order of RMF
`outfil` – output RMF name

Chapter 9

An OM Data Processing and Analysis Primer

As with EPIC and RGS datasets, many files are associated with an OM dataset. The `INDEX.HTM` file, and links therein, are viewable with a web browser and will help you navigate the dataset. The different types of files are discussed in Chapter 5.2; however, since the OM is somewhat different from the other instruments on-board XMM-Newton, we will discuss them in more detail in later sections.

The OM can operate in IMAGING, FAST, and GRISM mode. Each of these modes has dedicated commands to reprocess the data: *omichain*, *omfchain*, and *omgchain*. These are Perl scripts which each call several procedures sequentially that are used to prepare the data for processing, make and apply flatfield images, and detect sources. The tasks *omichain* and *omfchain* also calculate the instrumental magnitudes of sources, find the position of the sources (in equatorial coordinates), and produce a sky image; *omgchain* produces a spectrum. If you run these chains, it is helpful to inspect the `sas_log` file to get a detailed list of the performed tasks. These chains rely on filters specified by the user; if no arguments are given, they run on all the files present in the ODF directory. **Due to the long file names and the large number of input parameters, users are urged to simply use the chains and not run the chains' individual tasks one at a time.**

Most OM data are obtained in IMAGING mode. If they were obtained in the FAST mode, there will be an additional event list file corresponding to the Fast window (`*FAE.FIT`). Reprocessing of data taken in FAST mode is discussed in §9.3. Reprocessing OM GRISM data is discussed in §9.4.

At this point, it is assumed that you have downloaded the data from the HEASARC archive onto a Hera server, standard or anonymous Hera is running (see §4.2), you have prepared the data for processing (see Chapter 6), and the working directory PROC has been made. For example data, we will use observations of the Lockman Hole (Obs ID 0123700101; the same as for the EPIC walk-through) for Image mode, Mkn 421 (Obs ID 0411081601) for Fast mode, and BPM 16274 (Obs ID 0125320801) for Grism mode, though any dataset with the appropriate mode will suffice.

9.1 OM Artifacts and General Information

Before proceeding with the pipeline, it is appropriate to discuss the artifacts that often affect OM images. These can affect the accuracy of a measurement by, for example, increasing the background level. Some of these can be seen in Fig. 9.1.

- Stray light – background celestial light is reflected by the OM detector housing onto the center on the OM field of view, producing a circular area of high background. This can also produce looping structures and long streaks.
- Modulo 8 noise – In the raw images, a modulo 8 pattern arises from imperfections in the event centroiding algorithm in the OM electronics. This is removed during image processing.
- Smoke rings – light from bright sources is reflected from the entrance window back on the detector, producing faint rings located radially away from the center of the field of view.
- Out-of-time events – sources with count rates of several tens of counts/sec show a strip of events along the readout direction, corresponding to photons that arrived while the detector was being read out.

Further, artifacts also can contaminate grism data. Due to this mode's complexity, users are urged to be very careful when working with grism data, and should refer to the SOC's website on this topic.

Users should also keep in mind some differences between OM data and X-ray data. Unlike EPIC and RGS, there are no good time intervals (GTIs) in OM data; an entire exposure is either kept or rejected. Also, OM exposures only provide direct energy information when in grism mode, and the flat field response of the detector is assumed to be unity.

For detailed descriptions of PP data nomenclature, file contents, and which tasks can be used to view them, see Tables 5.2 and 5.3.

If you simply want a quick look at your data, sky images and source lists are in `*SIMAGE*.FTZ` and `*SWSRLI*.FTZ`, respectively. Further, there are low resolution sky images for each filter; they follow the nomenclature:

`PjjjjjjkkkkOMX000RSIMAGbb000.QQQ`

- `jjjjjj` – Proposal number
- `kkkk` – Observation ID
- `b` – Filter keyword: B, V, U, M (UVM2), L (UVW1) and S (UVW2)
- `QQQ` – File type (e.g., PNG, FTZ)

So for example, `P0123700101OMX000RSIMAGV000.FTZ` is the final low resolution sky image in the V filter. To see what files have been summed to make the final image, search for the keyword `XPROC0` in the FITS header. For our example image, this would be

```
XPROC0  = 'ommosaic imagesets=''product/P0123700101OMS004SIMAGE1000.FIT produc&'
CONTINUE  't/P0123700101OMS415SIMAGE1000.FIT product/P0123700101OMS416SIMAGE10&'
CONTINUE  '00.FIT product/P0123700101OMS417SIMAGE1000.FIT product/P0123700101O&'
CONTINUE  'MS418SIMAGE1000.FIT'' mosaicedset=product/P0123700101OMX000RSIMAGV0&'
CONTINUE  '00.FIT exposuremap=no exposure=1000 # (ommosaic-1.11.7) [xmmsas_200&'
CONTINUE  '61026_1802-6.6.0]'
```

The source list file (`*SWSRLI*.FTZ`) also contains useful information for the user; the column names are listed in Table 9.1.

The source list file (`*SWSRLI*.FTZ`) also contains useful information for the user; the column names are listed in Table 9.1.

Table 9.1: Some of the important columns in the source list file.

Column name	Contents
SRCNUM	Source number
RA	RA of the detected source
DEC	Dec of the detected source
POSERR	Positional uncertainty
RATE	extracted count rate
RATE_ERR	error estimate on the count rate
SIGNIFICANCE	Significance of the detection (in σ)
MAG	Brightness of the source in magnitude
MAGERR	uncertainty on the magnitude

9.2 Imaging Mode

9.2.1 Rerunning the Pipeline

Please note that calling any of the repipelining tasks will initiate processing on all OM data of that particular mode; currently, only *omichain* will accept parameters to limit processing to a specific filter or exposure.

To rerun the pipeline on all exposures and filters, in the new Command Window you made at the end of §6, run the task:

```
omichain
```

This produces numerous files, including images and regions for each exposure and each filter. If we are interested in the sources detected in the mosaicked, V band image we could run *omichain* with the appropriate flags by typing

```
omichain filters=V processmosaicedimages=yes omdetectnsigma=2.0
   omdetectminsignificance=3.0
```

where

`filters` – list of filters to be processed
`processmosaicedimages` – process the mosaicked sky images?
`omdetectnsigma` – number of σ above background required for a pixel to be considered part of a source
`omdetectminsignificance` – minimum significance of a source to be included in the source list file

The output files can be used immediately for analysis, though users are strongly urged to examine the output for consistancy first (see §9.2.2). The chains apply all necessary corrections, so no further processing or filtering needs to be done. Please note that the chains do not produce output files with exactly the same names as those in the PPS directory (they also produce some files which are not included in the PPS directory at all.) Table 9.2 lists the file ID equivalences between repipelined and PPS files.

9.2.2 Verifying the Output

While the output from the chains is ready for analysis, OM does have some peculiarities, as discussed in §9.1. While these usually have only an aesthetic effect, they can also affect source brightness measurements, since they increase the background. **In light of this, users are strongly encouraged to verify the consistency of the data prior to analysis.** There are a few ways to do this. Users can examine the combined source list with *fv*, which will let them see if interesting sources have been detected in all the filters where they are visible. Users can also overlay the image source list on to the sky image with *implot* by downloading the relevant files to the user's local machine. This can also be done with *ds9* or *gaia* by using *slconv* to change source lists into region files and downloading the relevant files to your local machine. The task *slconv* allows users to set the regions radii in arcseconds to a constant value or scale them to header keywords, such as RATE. By default, *ds9* region files have suffixes of `.reg`; *gaia* region files have suffixes of `.gaia`.
In the following example, we make a region file from the source list for the mosaicked, V-band sky image by typing

```
slconv srclisttab=P0123700101OMS000RSISWSV.FIT radiusexpression=5
   outfileprefix=Vband_mosaic outputstyle=ds9
```

where

`srclisttab` – source list file name
`radiusexpression` – constant or expression (possibly involving keywords) used
  to determine the radii of the plotted circles
`outputstyle` – output format; either ds9 or gaia
`outfileprefix` – prefix of output file name

Table 9.2: File ID equivalences between repipelined and PPS OM files.

Repipelined Name	PPS Name	Description
EVLIST	none	FAST mode events list
FIMAG_	FIMAG_	combined full-frame image
FLAFLD	none	flatfield
FSIMAG	FSIMAG	combined full-frame sky image
HSIMAG	HSIMAG	full-frame HIRES sky image mosaic
IMAGE_	IMAGE_	image from any filter or GRISM
IMAGE_	IMAGEF	FAST mode image
LSIMAG	LSIMAG	full-frame LORES sky image mosaic
OBSMLI	OBSMLI	combined observation source list
REGION	SWSREG	sources region file
REGION	SFSREG	FAST mode sources region file
REGION	SGSREG	GRISM *ds9* regions
RIMAGE	GIMAGE	GRISM rotated image
RSIMAG	RSIMAG	default mode sky mosaic
SIMAGE	SIMAGE	sky aligned image
SIMAGE	SIMAGF	FAST mode sky aligned image
SIMAGE	none	GRISM sky aligned image
SPCREG	SPCREG	GRISM *ds9* spectrum regions
SPECLI	SPECLI	GRISM specra list
SPECTR	SPECTR	source extracted spectra
SUMMAR	SUMMAR	observation summary
SWSRLI	SWSRLI	sources list
SWSRLI	SFSRLI	FAST mode sources list
SWSRLI	SGSRLI	GRISM sources list
TIMESR	TIMESR	FAST mode source timeseries
TSHPLT	TSHPLT	tracking history plot
TSTRTS	TSTRTS	tracking star timeseries

The image and region files can be easily downloaded to a local machine for examination by dragging-and-dropping into a target directory. The mosaicked, V-band sky image, `P0123700101OMS000RSIMAGV.FIT`, with the region file from *slconv* overlayed, is shown in Fig. 9.1. There clearly are spurious detections; these can be removed by hand, or by rerunning *omichain* with a different background-level threshold or source significance.

9.3 Fast Mode

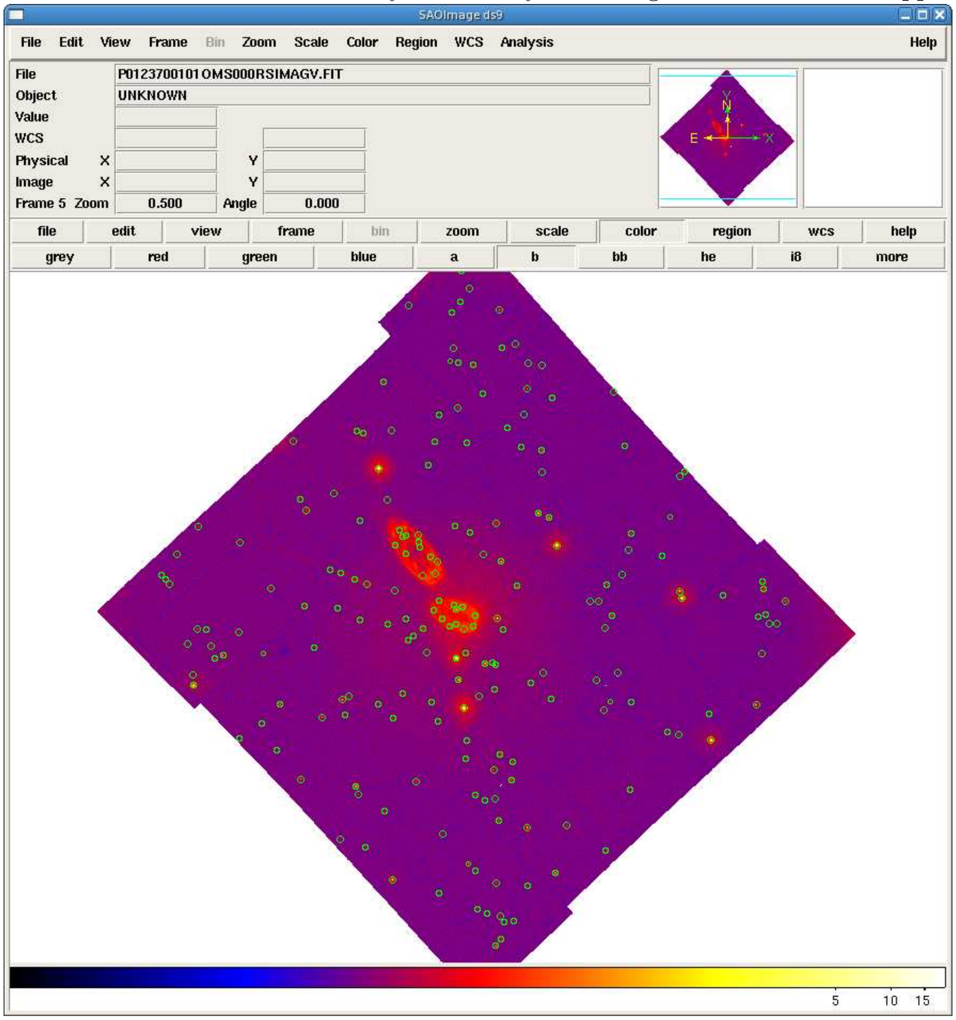
9.3.1 Rerunning the Pipeline

The repipelining task for OM data taken in fast mode is *omfchain*. It produces images of the detected sources, extracts events related to the sources and the background, and extracts the corresponding light curves. At present, unlike *omichain*, *omfchain* does not allow for keywords to specify filters or exposures; calling this task will process all fast mode data.

To run the pipeline on fast mode data, type

```
omfchain
```

Figure 9.1: A mosaicked, V-band sky image and corresponding region file viewed with *ds9*. Spurious detections are clearly present and can be removed either by hand or by rerunning omichain with the appropriate flags.



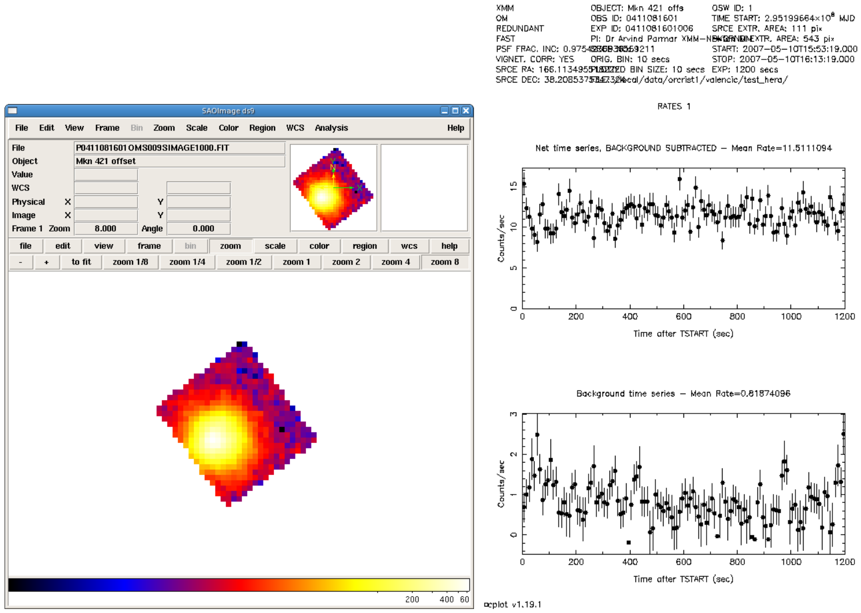
There are two types of output files: those that start with `f` are intermediate images or time series files; those that start with `p` are products. The output files are are described in Table 9.2.

To demonstrate some of these output files, we have rerun the pipeline on the example dataset and downloaded the output to a local machine. The processed image in sky-coordinates from one exposure, `P0411081601OMS006SIMAGE1000.FIT`, is shown in Fig. 9.2 (left). The background-subtracted light curve produced automatically by the task, `F0411081601OMS006TIMESR1000.PS`, is shown in Fig. 9.2 (right).

9.3.2 Verifying the Output

A good first step in checking the output is to examine the light curve plot for both the source and background, making sure they are reasonable: no isolated, unusually high (or low) values, and no frequent drop-outs. Users should also check the image with *fv* or *ds9* in the Fast mode window to see if the source is near an edge. If it is, it's a good idea to examine the light curves from diffent exposures to verify that they are consistent from exposure to exposure (while keeping in mind any intrinsic source variability). If the image is blurred or unusual in any way, users should check the tracking history file to verify the tracking was reliable.

Figure 9.2: Left: The processed FAST mode sky image. Right: the light curve produced automatically by *omfchain*.



9.4 Grism Analysis

9.4.1 Rerunning the Pipeline

The repipelining task for OM data taken in grism mode is *omgchain*. It produces images of the detected sources and background, extracts source spectra and region files, and makes source lists and postscript and PDF plots. At present, unlike *omichain*, *omgchain* does not allow for keywords to specify filters or exposures; calling this task will process all grism mode data.

To run the pipeline on grism mode data, type

```
omgchain
```

There are two types of output files: those that start with `g` are intermediate or auxiliary files and source lists; those that start with `p` are products. The output files are are described in Table 9.2.

To demonstrate some of these output files, we have rerun the pipeline on the example dataset and downloaded the output to a local machine. The processed image, rotated to align with the columns of the image (`p0125320801OMS005RIMAGE0000.FIT`), is shown in Fig. 9.3 (left). Two region files are overlayed: `p0125320801OMS005REGION0001.ASC`, which corresponds to the sources detected in this rotated image (green), and `p0125320801OMS005SPCREG0001.ASC`, which corresponds to the sources in the spectra list file (red) and indicates the locations of the zero and first orders. The task *omgchain* automatically extracted the spectrum of the red region (`p0125320801OMS005SPECTR0000.FIT`); this is shown in Fig. 9.3 (right).

9.4.2 Verifying the Output

The correct correlation of zero and first orders is crucial for grism analysis. Users should inspect the rotated image with *fv* or *ds9* and verify the identification of the orders by overlaying the `*SPCREG*` region file, as shown in Fig. 9.3 (left); the `*SPECLI*` file also contains this information. If users are interested in all source detections, the region file can also be overlayed and the full source list examined. Users should also examine the spectra plots automatically produced by *omgchain*, for both the source and background, making sure they are reasonable. For improved source detection, the parameter `nsigma` can be changed.

Figure 9.3: Left: The repipelined, rotated image with regions overlayed. Right: the spectrum extracted from the source (red region).